Copy No. _____

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Utilizing Arc Marine concepts for designing a geospatially enabled database to support rapid environmental assessment

Anthony W. Isenor
Defence R&D Canada – Atlantic

Tobias W. Spears
Fisheries and Oceans Canada
Bedford Institute of Oceanography

Defence R&D Canada – Atlantic

Technical Memorandum
DRDC Atlantic TM 2009-061
July 2009

Canada

This page intentionally left blank.

Utilizing Arc Marine concepts for designing a geospatially enabled database to support rapid environmental assessment

Anthony W. Isenor
Defence R&D Canada – Atlantic

Tobias W. Spears
Fisheries and Oceans Canada
Bedford Institute of Oceanography

Defence R&D Canada – Atlantic

Technical Memorandum
DRDC Atlantic TM 2009-061
July 2009

Principal Author

Original signed by Anthony W. Isenor

Anthony W. Isenor

Defence Scientist

Approved by

Original signed by Francine Desharnais

Francine Desharnais

Head, Maritime Information and Combat Systems Section

Approved for release by

Original signed by Ron Kuwahara

Calvin Hyatt

DRP Chair

Abstract

The design of the rapid environmental assessment (REA) database version 1 was completed under contract. The database was constructed in PostgreSQL, an open-source database management system. The REA database was primarily used for the storage of DRDC Atlantic environmental data. However, additional data sets from external sources were added for bathymetry and geological data. As use of the REA database increased, it became desirable to redesign the database to better serve the user community within DRDC Atlantic. The redesign effort focused on the use of widely used standards and specifications for oceanographic data and metadata management. The redesign created a complete data model in the ERwin data modelling software for a production level database. The data model is fully documented in terms of data type, comment fields and relationships between entities. The redesign effort also fully documented the mapping of data from the existing REA database to the redesigned production data model, thereby providing developers with a clear and concise progression plan. The redesign effort also identified considerable data in the existing REA database that is not required in the production database. Finally, a designed data classification scheme is used to develop a user exit strategy for accessing the external data sets. This negates the need to store external data sets within the redesigned database.

Résumé

La conception de la version 1 de la base de données d'évaluation environnementale rapide (EER) a été réalisée en vertu d'un contrat. La base de données a été élaborée à l'aide du PostgreSQL, un système de gestion de base de données à code source ouvert. La base de données EER a servi principalement au stockage de données environnementales de RDDC Atlantique. Toutefois, des jeux de données supplémentaires provenant de sources externes et contenant des données bathymétriques et géologiques se sont ajoutés à la base de données. Au fur et à mesure que la base de données EER augmentait, il devenait souhaitable de revoir la conception de la base de données afin de mieux desservir la communauté des utilisateurs à RDDC Atlantique. Cette activité de révision a porté sur l'utilisation de normes et de spécifications couramment utilisées pour la gestion de données et de métadonnées océanographiques. La révision a permis de créer un modèle de données complet à l'aide du logiciel de modélisation de données ERwin pour obtenir une base de données de niveau de production. Le modèle de données est entièrement documenté, en termes de type de données, de champs de commentaires et de relations entre les entités. L'activité de révision a également permis de pleinement documenter la mise en correspondance des données, de la base de données EER existante au modèle révisé de production, fournissant ainsi aux développeurs un plan de progression clair et concis. En outre, la révision a permis de relever, dans la base de données EER existante, des quantités considérables de données qui n'étaient pas nécessaires dans la base de données de production. Enfin, un système de classification des données est utilisé pour mettre au point une stratégie d'exit utilisateur permettant d'accéder à des jeux de données externes. Ainsi, il devient inutile de stocker des jeux de données externes dans la base de données révisée.

This page intentionally left blank.

Executive summary

Utilizing Arc Marine concepts for designing a geospatially enabled database to support rapid environmental assessment

Anthony W. Isenor and Tobias W. Spears; DRDC Atlantic TM 2009-061; Defence R&D Canada – Atlantic; July 2009.

Background: The REA database was primarily used for the storage of DRDC Atlantic environmental data. As use of the REA database increased, it became desirable to redesign the database to better serve the user community within DRDC Atlantic. The redesign effort focused on the use of widely used standards and specifications for oceanographic data and metadata management. The redesign created a complete and highly documented data model including full documentation on the mapping of data from the existing REA database to the redesigned production data model.

Results: The REA data model clearly documents the management solution for those types of data common to collection exercises at DRDC Atlantic. The model utilizes standards and specifications in the oceanographic community for both the data and metadata components. For the data, we have shown the utility of the Arc Marine data model. For metadata, we have utilized components of the International Organization for Standardization (ISO) 19115 standard for geospatial metadata. The resulting data model fully supports all environmental data types that are common to oceanographic surveys and provides a scalable and flexible strategy for incorporating external data sets into the REA system.

Significance: The utilization of standards or specifications during the data model design provides an enhanced level of compliance with other similar organizations. This means the results of this effort have the potential to improve interoperability between producers and users of REA data. This could influence design strategies and developments associated with environmental databases for groups such as DND Meteorological and Oceanographic (MetOc) Office and Fisheries and Oceans sections dealing with geospatial enabled oceanographic data sets. The completed data model also represents an essential information component of a larger REA system and the utilization of open-source software provides cost-benefit gains. As well, the developed data model could easily be transferred to an operational system and integrated with models that deliver REA operational products to deployed forces.

Future plans: Plans include the assessment of the data model by users and developers within DRDC Atlantic. If the model proves acceptable, data porting from the existing database to the new database will be conducted.

Sommaire

Utilizing Arc Marine concepts for designing a geospatially enabled database to support rapid environmental assessment

Anthony W. Isenor and Tobias W. Spears; DRDC Atlantic TM 2009-061; R & D pour la défense Canada – Atlantique; Juillet 2009.

Introduction ou contexte : La base de données EER a servi principalement au stockage de données environnementales de RDDC Atlantique. Au fur et à mesure que la base de données EER augmentait, il devenait souhaitable de revoir la conception de la base de données afin de mieux desservir la communauté des utilisateurs à RDDC Atlantique. Cette activité de révision a porté sur l'utilisation de normes et de spécifications couramment utilisées pour la gestion de données et de métadonnées océanographiques. L'activité de révision a également permis de créer un modèle de données complet et très bien documenté, comprenant notamment une documentation complète sur la mise en correspondance des données, de la base de données EER existante au modèle révisé de données de production.

Résultats : Le modèle de données EER documente clairement la solution de gestion pour les types de données utilisés couramment pour des activités de collecte à RDDC Atlantique. Le modèle utilise des normes et des spécifications en usage dans le milieu de l'océanographie tant pour les éléments de données que pour les éléments de métadonnées. Pour les données, nous avons démontré l'utilité du modèle de données Arc Marine. Dans le cas des métadonnées, nous avons utilisé des éléments de la norme 19115 de l'Organisation internationale de normalisation (ISO) relative aux métadonnées géospatiales. Le modèle de données qui en résulte accepte tous les types de données environnementales utilisés couramment dans les levés océanographiques et permet une stratégie extensible et souple pour intégrer des jeux de données externes dans le système EER.

Importance: L'utilisation de normes et de spécifications au cours de l'élaboration du modèle de données assure un meilleur niveau de conformité avec d'autres organisations similaires. Par conséquent, les résultats des efforts déployés en ce sens offrent la possibilité d'améliorer l'interopérabilité entre les producteurs et les utilisateurs des données EER. Cela pourrait influencer les stratégies de conception ainsi que l'élaboration, associées aux bases de données environnementales pour des groupes tels que le Centre météorologique et océanographique (MetOc) du MDN et les sections de Pêches et Océans qui s'occupent de jeux de données océanographiques à référence géospatiale. Le modèle de données obtenu représente également un volet d'information essentiel dans un système EER plus élaboré, et l'utilisation d'un logiciel à code source ouvert offre des avantages en ce qui concerne la rentabilité. En outre, il serait facile de transférer le modèle de données vers un système opérationnel et de l'intégrer à des modèles qui permettent de livrer des produits EER opérationnels aux forces déployées.

Perspectives : Les projets comprennent l'évaluation du modèle de données par les utilisateurs et les développeurs au sein de RDDC Atlantique. Si le modèle s'avère acceptable, le portage des données sera effectué depuis la base de données existante vers la nouvelle base de données.

Table of contents

List of figures

List of tables

Acknowledgements

The authors would like to acknowledge and thank Fisheries and Oceans, Science Branch, Maritimes Region, Ocean Sciences Division for recognizing the importance of this interdepartmental effort and supporting the agreement which allowed the effort to proceed. We also thank the Harmonized Model Management Group (HMMG), especially Dr. John Herring and Mr. Shawn Silkensen for supplying the unified Rational Rose model of ISO 19115. This was instrumental in helping us navigate ISO 19115 and the suite of supporting ISO specifications. We also thank Herman Varma, DFO, for his expertise and ongoing support in working with ISO 19115 and geographic data management. Finally, we acknowledge the Arc Marine working group for providing the framework for the REA database design.

This page intentionally left blank.

1 Introduction

Knowledge of your environment is important to many of your activities. The environment in which you move and function can have significant influence on your ability to carry out your daily routine. This applies to the individual involved in a typical daily activity or a group working towards a common goal.

For the Canadian Forces (CF) and in particular the Canadian navy, the environment (i.e., in the sense of environmental rather than office or work space) of operation is dominated by two physical parts: the ocean and the atmosphere. Both parts have an enormous influence on naval operations and by understanding the processes that operate within these parts we are better prepared to successfully conduct operations. The collection and analysis of environmental data helps us better understand our operating environment and whether or not changes in our environment are important to the specific naval operation.

As a naval research centre, DRDC Atlantic has a long history of data collection in the marine environment. Longard (1993) described the centre's origin in the 1940's with ocean data collection activities present from the beginning. A considerable amount of data has been collected over this history. Typically, these data have been collected and used on an individual at-sea experiment. After returning to shore, the data would be stored or archived in some form, often related to the originating system or sensor. There was very little effort to combine different data sets or different data types from multiple experiments.

The original data collection activities produced data sets that were stored on paper records. With the advent of computers, the data became digital and storage moved from paper to computer files and directories.

The file and directory structure was used for data storage well into the 21st century. Although this provided a rudimentary means to track and account for data sets, the process lacked several features that are more common to a managed system. First, there is no common access method(s) for acquiring the data. Second, there is no means for quick discovery of data within a defined region of space and time. Third, there is no process control which can track the history of data manipulation. Finally, there is no common assignment of computer typing based on data type.

All of these factors limit the data's usability. For example, no common access methods means users must acquire specialized knowledge specific to a data set before they can utilize those data. Having no discovery methods, means the user must acquire knowledge on the spatial-temporal aspect of the data from personal experience or mass plotting of data positions. The lack of process control means the errors detected and corrected by previous users of the data set do not stream back to the original data, but rather remain in the data set for other users to (hopefully) find and correct. Finally, no common assignment of typing means data of similar types are not treated similarly. As an example, profile data based on temperature or salinity may not be treated in similar ways.

The combination of these factors also limits the functionality. For example, the requirement for specialized access methods results from similar data sets being treated differently. Each of these data sets then requires specific and specialized methods for acquiring the data. Consider this

factor combined with the lack of process control. Process control could be enhanced by simply including qualify flags on the data. However, adding quality flags to each specialized data set requires the modification of each specialized access method. Thus, multiple modifications are required for implementation of the quality flags. Often this means the addition is infeasible in terms of compliance across all data sets and also considering long term maintenance of the software components. Effectively, the multitude of individual systems limits the functional scalability of the system as a whole.

1.1 Rapid environmental assessment

The functional scalability noted above is referring to aspects of the data system that are related to the system's data analysis or data manipulation capabilities. This is a very system-driven view and for both ourselves and the navy, we must remain cognisant of the larger objectives, specifically the successful completion of a sea trial or naval operation. Such objectives are actually the drivers that dictate any level of effort placed on modifications to the data system. Alternately stated, it is the decision requirement of the trial or operation that must dictate modification to the data system.

In an operational setting, the navy is often interested in how the particular operating environment impacts their available sensors and weapons. In the underwater domain, the primary concern is related to the transmission of sound through the water, and how the performance of sonar equipment is impacted by variations in sound propagation and more generally, by variations in environmental conditions (Chapman, *et al.* (1997), Hutt, *et al.* (2002), Osler, *et al.* (2002)). To concentrate efforts towards assembling the data necessary for such assessments, DRDC Atlantic initiated the Rapid Environmental Assessment (REA) Applied Research Project (ARP) in 2004. This project aimed to provide the anti-submarine warfare commander with key environmental data to be used for the assessment of sonar performance. Since the data were to be utilized during tactical situations, the rapid nature of the environmental assessment was necessary. As a result, the database created for the project became known as the REA database.

One outcome of this ARP was a survey conducted by Whitehouse (2004) which identified research and development opportunities related to REA. The survey examined the unclassified literature related to REA and also solicited views and options from CF personnel, DRDC Atlantic staff, and international experts in the REA community. The survey identified enabling technologies for REA such as the internet and geographic information system (GIS) technologies, and numerical modelling for now-casting and adaptive sampling strategies. Access to numerical model output for now-casting and adaptive sampling is particularly important in the littoral zone, where there exist short spatial and temporal scales for many important oceanographic variables.

Consistent with the identified opportunities, DRDC Atlantic utilized the ARP to begin the process of developing a data management system for its sea trial environmental data. This activity began by focusing on three primary environmental data sets collected by the Centre:

- temperature profiles collected by eXpendable BathyThermographs (XBT),

- an assortment of oceanographic and atmospheric environmental data, including ship propulsion data from the CFAV QUEST Non Acoustic Data Acquisition System (NADAS), and
- acoustic transmission loss data.

These historic data sets were archived in file-based systems. In some cases, data were non-quality controlled, direct from source, data files. Effectively, there was storage of the data but no real management of the data. This aspect of the REA project was designed to produce a management system for the Centre's environmental data.

The construction of REA database version 1 (hereafter known as REA load database) was loosely specified around an open-source database management system (DBMS) known as PostgreSQL. The data were recognized as being largely geospatial and thus a GIS was considered the most reasonable way to proceed. The Postgre suite of software also had PostGIS which would address this requirement.

The REA load database was constructed via a contract issued for the design of the database and for the loading of the Centre's existing data into the database. The process of database design is known as data modelling. The design aspect can be thought of in the more common architectural sense, similar to designing a building. In this case, the design applies to the table and field structure of the database. As with architectural design, there is seldom one clear answer to a specific data modelling problem. There will be multiple designs that can meet all the stated requirements of a particular system. The designs simply apply trade-offs between aspects of functionality (e.g., scalability, ease of maintenance, database performance).

The initial contract activity to design, build and load the database resulted in what DRDC Atlantic staff refer to as the REA DB (i.e., the REA database). Various design decisions resulted in a highly specialized table structure which lacked flexibility and conformity to existing marine and geospatial standards. For these and other reasons (documented in later sections), the decision was made to redesign the database. This report outlines the redesign effort.

1.2 Report outline

This report represents the output of the REA database redesign effort. This report extensively discusses the relationships between the input data and the database structure. As well, we document the table and field structure of the existing database. This is required to produce a level of understanding as to the scope of the existing database. This understanding is needed when moving the data from the existing database to the redesigned database.

An important distinction is drawn between the existing REA database and the database to be constructed from the data modelling conducted in this study. The existing database has become known as the REA DB, indicating a database that supports the REA activity. The existing database also supports the initial data loads and in many cases table structures were specifically

designed to support the loading from source data files. At present, the existing DB is at version 3 beta and it is this version that is documented in this report.

For this report the existing database will be referred to as the REA load database (REA LDB). Referring to the existing DB as the REA LDB indicates a shift in the intended functionality. The initial REA DB was constructed to support both the loading of data and the functions that support rapid environmental assessment. In contrast, a LDB is designed to support the initial loading of data into the DB and the correction of data conflicts during the load process. As a result, the LDB has specific table structures more consistent with the initial data sets. A load DB also does not typically support the primary function associated with the initial intent of the project.

The work presented here has resulted in a redesigned data model that will support the creation of a new REA database. This new database will be referred to as the REA production database (REA PDB). A PDB typically has more generalized data structures and supports the primary intent of the project, in this case rapid environmental assessment, while not dealing with data issues related to the initial ingest of raw data sets.

Finally, the entire system that is being constructed is referred to as the REA system; or REAS. The redesign effort has taken a system view, not a database view. In that regard, the redesign has replaced database complexity with data classifications schemes. This means the data system has a database (i.e., the REA PDB) as one component of the system, but not as the only component. As well, it means complexities within the LDB schema that handled specific data input forms have been replaced by procedures that classify the input data and address common input types in common ways. This provides the REAS with much greater scalability.

This report first provides background information on the concept of data modelling. Next, we examine the existing REA LDB and discuss what we perceive as deficiencies with the LDB. We then introduce database structure used in many oceanographic database applications. This structure, as described by Wright, *et al.* (2007), is known as Arc Marine. Arc Marine was developed for the ESRI (2009) geodatabase environment but is applicable for this Postgre-based system.

We then introduce and discuss spatial reference systems. In many respects, the spatial reference system is the forgotten complication to geospatial data systems. This is especially true when dealing with older data sets that lack explicit information on the spatial reference system used during collection.

We then detail the actual design of the REA PDB. These report sections consider individual data types and detail how these data are placed within the PDB. We also provide mappings from the LDB structure to the PDB structure. We discuss some complexities of the data collection activities, including the Non Acoustic Data Acquisition System. It is important to understand the data acquisition before attempting to understand the specifics of the data themselves. This design discussion also includes vertical profile data, the east coast ambient noise data set, sediment data, transmission loss data, and gridded data sets.

The REAS concept is then developed. By taking a system view, we eliminate the need to address the data modelling aspect for the bathymetry and bottom types (e.g., geoclutter) data sets that currently exist in the REA LDB. Neither of these data sets originated with DRDC Atlantic and as

such, DRDC Atlantic holds no responsibility for the long-term archival of these data. As well, the data may have new releases in the sense that new bathymetric products or new surficial geology products could become available at any time. Thus, the REAS needs to be capable of easily handling additions or updates to products similar to the bathymetry and surficial geology. By not placing these products directly into the PDB, we provide the capability to use the products without the enhancement and maintenance costs associated with direct storage inside the PDB. This functionality uses the concept of user exits.

Various business processes for REAS are also introduced. These processes outline how to incorporate a new data set into the REAS, the table design process for adding new tables, the lineage processing steps, and data extraction for sea trial planning. These business processes are outlined in very general terms.

Finally, we present a component of the data model for handling the lineage of the data sets. Here, lineage refers to the processing provenance or processing history associated with a particular data set within the REAS. The lineage component of the data model allows the tracking of the data processing, those responsible for that processing, and the citations to the publications related to the data sets. The lineage component of the data model is based on the International Organization for Standardization (ISO) standard for geospatial metadata, known as ISO (2003) 19115.

1.3 Nomenclature

1.3.1 For readers of this report

Throughout the text there will be references made to database table and field names. We attempt to provide clarity for the reader by differentiating table and field names used within the data model. As a means to accomplish this, we identify table names using the Arial font, and field names using italic Times Roman. As an example, the table Measurement_Location contains a field named *Feature_Code*.

We also provide a Glossary at the back of this document. The Glossary lists the terminology used in this document and definitions for these terms.

1.3.2 For software developers

We also attempt to provide a readable report by using upper and lower case table and field names (as shown in above example) with name separation using an underscore. Although this provides readability, it is likely that the implemented database will not follow the upper and lower case naming. This is related to the data modelling environment used in this work, which is the Computer Associates (2009) ERwin software, version 4.1.2522. Using the data model, ERwin generates the structure query language (SQL) instructions that are necessary to create the database in PostgreSQL. However, PostgreSQL does not follow the upper and lower case characteristics of the naming without enclosing the name in quotation marks. Since the upper and lower case

naming adds complication when writing SQL commands to the database, we recommend that any instantiation of the data model NOT include the upper and lower case naming.

In a simplistic example, what this means is that a table created using the SQL command:

```
CREATE TABLE Measurement_Location;
```

can be assessed using the PostgreSQL command:

```
SELECT * FROM measurement_location;
```

Note the table can be accessed using all lowercase.

2 Data modelling background information

Before discussing data models or databases, we need to establish background information and the language to be used for the discussion. As well, we need to establish a collective understanding around the issue of data models and in particular, acknowledge that data modelling is a design process. As described by Simsion (2007, pg 12), the design process results in multiple solutions to the same data modelling problem.

The multiple solutions which result from the data modelling exercise have resulted in many problems both in civilian and military fields. In a military context, data models such as the Joint Consultation Command and Control Information Exchange Data Model (JC3IEDM), the Maritime Information Exchange Model (MIEM), the Universal Core data model (UCore), or the National Information Exchange Model (NIEM) claim to address similar requirements, but in some sense "better" than the others. Each development team considers their particular model to be better suited to the requirements. These conflicting claims simply confuse the user community and often don't result in any progression of the underlying issue of data use; after all it is the use of the data that is critical to addressing military needs. Instead, such claims amplify hostilities among the individual data model communities.

Diversity in data models is a recognized result of the data modelling activity. However, there are many factors which influence any resulting data model. These factors certainly include the requirements of the system being constructed. However, full requirements are often not specified. In such cases, the data modeller's knowledge of the business rules and data types will help guide, define, and shape the data model.

2.1 Defining data modelling

It is generally agreed that there are three types of data modelling: conceptual data modelling; logical data modelling, and; physical data modelling. It is also recognized that there is no consensus among academics or practitioners as to the boundaries between these different types of modelling (Simsion (2007)). However, for the sake of this work we follow the Simsion (2007) definitions and describe the three types of modelling as follows:

- conceptual data modelling: a database independent view of the data
- logical data model: converts a conceptual data model into a form that uses the data definition language of a specific database implementation.
- physical data model: converts the logical data model into an implementation for a specific DBMS, where alterations can be made to address performance issues.

Based on these definitions, this work has resulted in the construction of both conceptual and logical data models.

2.2 Quality in data modelling

The research of Simsion (2007) concluded that data modelling was a design activity. As with any design activity, the aspect of creative thinking results in design diversity. This diversity then introduces the problem of effectively and objectively comparing data models, and more specifically the objective assessment of data model quality.

Research into the quality of data models has been conducted by Moody and Shanks (1994), Moody, *et al.* (2003), Moody and Shanks (2003), Leung and Bolloju (2005), and Simsion (2007). The work of Moody and Shanks (2003) and Simsion (2007) is particularly applicable as these efforts attempted to isolate the important components of a framework for assessing data model quality.

A list of data model quality indicators was first proposed by Moody and Shanks (1994). Moody and Shanks (2003) extended this initial research by conducting a five year study that validated the main quality indicators. These indicators were revised and grouped according to four stakeholder groups. The four stakeholder groups and resulting eight quality indicators are:

Business User:

- completeness: refers to the data model containing all user requirements.
- integrity: refers to the proper definition of business rules within the data model.
- flexibility: refers to the ease with which the data model can cope with business or regulatory change.
- understandability: refers to the ease with which the data model concepts, structures, etc. can be understood.

Data Analyst

- correctness: refers to whether the data model conforms to the rules of the data modelling technique (i.e., is it a valid data model). This includes minimizing data redundancy. This indicator may be considered syntactic correctness.
- simplicity: refers to the model containing the minimum possible entities and relationships.

Data Administrator

- integration: refers to the consistency of the data model within the scope of the organizations other data assets.

Application Developer

- implementability: refers to the ease of implementation of the data model including such things as being implemented within time, on budget, and technology constraints.

The empirical research of Moody and Shanks (1994) indicated that the primary quality indicator (accounting for 50% of the quality variance) was understandability, followed by completeness at

36%, with the remaining 14% distributed among correctness (9%), simplicity (3%) and flexibility (2%). Both understandability and completeness are in the Business User group, indicating that users want data models that meet their particular business needs and are at a level they can understand. This also indicates the importance of identifying and knowing the customers of the database. Knowing your customers is seen as one of the important steps to delivering quality information (English (2002)).

The more recent Simsion (2007) research was oriented towards the goal of identifying the data modelling activity as either description or design. In this research, the quality factors were also identified. However, the work identified understandability as the smallest contributor to quality. Simsion (2007) considers this a possible artefact of the experimental design. The Simsion (2007, pg 259) work involved expert evaluation of 10 data models containing a mean of 11 entities and relationships. The Moody and Shanks (2003) evaluations involved 35 data models, with experts evaluating models containing a mean of 35 entities and relationships, and novices evaluating data models with a mean of 24 entities and relationships. In the 2003 study, the experts may have underrated the importance of understandability given the smaller size of the data models. In other words, the models being considered in the study may have already been considered "simple".

The completeness and correctness quality factors were not tested in the Simsion (2007, pg 231) study.

The initial simplicity quality factor was replaced by Simsion (2007, pg 259) with a complexity quality factor. Complexity, which is the opposite of simplicity, showed a positive correlation indicating that the expert evaluators typically correlated higher quality data models with higher entity and relationship counts. Note that this quality indicator is the only objectively measured indicator.

Finally, the flexibility indicator was found by Simsion (2007) to be the most important predictor of quality (opposite of Moody and Shanks (2003) findings). In the Simsion (2007) study, flexibility was determined by the expert evaluators based on the 5 point Likert scale (see Wikipedia (2009)). English (2006) actually proposes a division of the flexibility scale defined by Moody and Shanks (2003) into flexibility and stability. In this division, flexibility means the data model can support changes to business processes without major modification to the model, while stability means new applications (e.g., numerical models or tactical decision aids) can use the existing database directly or by simply adding tables to the database (but not modifying existing tables). In the Simsion (2007) study, the company where the case study originated actually implemented a data model that concentrated on flexibility rather than stability.

These results indicate inconsistencies in the research findings for data model quality metrics and the inability to adequately measure existing quality metrics. As well, the metrics are not independent, and this represents a serious deficiency. For example, correctness and understandability are linked. If a model lacks correctness, there could be cases where incorrectness results in lack of stakeholder understanding. Nevertheless, metrics remain a requirement of the process. If we consider the existing metrics and use the studies as indicators of importance, then the metrics of understandability, flexibility, and completeness may be considered important for a data model.

3 The existing REA LDB and associated design issues

The authors claim there are numerous deficiencies with the design of REA LDB. The underlying reasons or events which resulted in these deficiencies could be numerous. For example, inadequate time on the modelling task, inability of the customer to articulate requirements, or the disjointed efforts of those working to meet multiple deadlines on multiple projects. Nevertheless, we consider these deficiencies sufficiently important to require a rather substantive redesign effort. In the following section we detail these deficiencies.

3.1 Documentation

The initial work that created the REA LDB also resulted in the creation of a project final report by Deveau (2006). The report contained Annex A which included detailed output from the Enterprise Architect (Sparx Systems (2009)) data modelling software.

These Annex pages also contain the conceptual model diagrams which are useful to frame the extent of the data model. However, the information at an entity level is sometimes incomplete and often lacks content that is relevant to any of the stakeholder groups. This inadequacy results in the documentation being ignored during efforts such as process integration (i.e., the integration of a calculation or process to utilize the data contained in the database). In the formalism of the data model quality metrics, this lack of documentation produces an understandability issue.

3.2 Evolving design

One issue with the current REA LDB design is what we refer to as an evolving design. This refers to what English (2006) describes as a lack of flexibility when adding data of a particular geometric type, when that geometric type already exists in the LDB.

The geometric type is a concept related to the spatial-temporal characteristics of the data type. For example, a particular data type would be temperature, salinity, or sediment thickness. Particular geometric types could be vertical profiles or horizontal surfaces. In the case of temperature and salinity, a vertical profile of these data types is quite common. For temperature, such a vertical profile would be obtained from an XBT while a salinity profile could originate from a Conductivity-Temperature-Depth (CTD) sensor.

For the addition of data to the LDB when that geometric type already exists in the LDB, there should be few to zero additional tables and relationships required for the addition of the different data in the existing geometric type. For example, when the data model is capable of handling a temperature profile, it should be capable of handling another scalar data type expressed as a profile with little to no alterations in the data model. In the formalism of the quality metrics, the

inability of the model to incorporate different data types using the same geometric type is a completeness issue.

In the present case when the database structure is modified to accommodate the data, the modification to the structure may indicate that the data classes of the organization were never fully explained or understood. If this is the case, the specification of initial requirements for the database may be at fault. This would not be atypical, as studies have shown that about 70% of reported system defects were the result of undocumented or incorrectly documented requirements (see Lauesen and Vinter (2000)). As well, incomplete requirements are the most often cause of cancelled developments (Standish Group (1995)).

3.3 Lack of DBMS utilization

A DBMS is a software package that provides the capability to create and maintain a database. A DBMS follows basic rules as defined in the SQL specification (as described in Section 4.1.1). Utilization of the DBMS functionality is important for data integrity and consistency within the database.

The present REA LDB design does not utilize much of the functionality of the DBMS in which it is operating. The REA LDB design lacks in two important areas:

1) The data typing within the REA LDB is inconsistent

This means the syntactic type assigned to the data values is inconsistently applied for the same data values in different tables. As an example, consider the typing assigned to depth fields on table tsd_geopoints (i.e., depth field is type float(8)), the depth field in bathy_lines (i.e., depth field is type float(4) or real), and the depth field in xbt_file_meta_data (i.e., depth field is type varchar). Such variation makes porting the data to other databases a more difficult task, as each type needs to be identified and mapped to a type in another database. As well, it makes the utilization of the data by numerical models an overly complicated task, as each access needs to accommodate diverse syntax for the same data type. In the formalism of the quality metrics, this is a correctness issue.

2) Referential integrity constraints are often not applied

This means data integrity suffers or data integrity must be dealt with at the business process or worse, at the consumer level. In the formalism of the quality metrics, this is an integrity issue.

The second issue is common in GIS applications. In fact, many of the Arc Marine (Wright, *et al.* (2007)) GIS applications examined during this effort showed a lack of referential integrity utilization at the DBMS level. It is recognized that the ESRI implementation of a geodatabase does not utilize DBMS referential integrity constraints. More on this topic in section 4.1.1.

3.4 Lack of functionality

The present REA LDB design does not have the functionality that is required for the applications that will be accessing the data. For example, the model does not presently include quality flags on the data values. As well, there is no ability to query metadata (Stocks, Neiswender, *et al.* (2009)) associated with the data or instrumentation.

In terms of quality flags, the ability to store a measurement quality flag is essential to maintaining high quality data within a database and also to benefiting from previous analyses that identify quality issues specific to individual datum values. In the formalism of the data model quality metrics, this is a completeness issue.

The present design could be modified to add quality flags to the tables containing measurements. However, the current design would require composite codes (i.e., one quality code which applies to multiple data values in different data types) or more likely, individual quality control fields for each individual measurement field. If additional measurements are made for data types not currently within the data class, then additional fields for both the data values and quality flags would be required. This again leads to an evolving design. This issue is related to stability as described by English (2006).

The clarity of some metadata values is also lacking. Functionally speaking, the metadata are not easily accessible if an application requires those metadata. For example, the XBT coefficient values are not currently stored as numbers, and also are not currently stored in a consistent manner. As well, the two coefficients used in the processing of the XBT data are currently stored in two of three possible table fields, the exact arrangement depending on the coefficient naming in the input files. This naming has no bearing on how the coefficients are used in the processing. Since these coefficients are critical to the processed temperature profile (Hallock and Teague (1992), Kezele and Friesen (1993), Hanawa, *et al.* (1995)), a full review of these coefficients is recommended.

3.5 Not scalable

The extensibility or expandability for external data sets is not scalable in the present design. This means we can't easily continue to add external data to the REA system without ultimately hitting a data volume that we cannot maintain or process.

The current practice is to add data from external sources directly into the REA LDB. This creates two problems:

1) when the external data is updated, the REA LDB needs to be purged of these external data and reloaded with the new external data.

2) the volume of external data can easily exceed our capacity. We cannot hope to import all external data of relevance to the DRDC Atlantic marine area of interest without having serious disk usage issues.

To some extent, this was an ill-conceived requirement of the initial design specification. For example, the actual requirement was for a design that allowed portability of the REA system to multiple platforms. Although the majority of database related processing would be done while at-shore, there was a need for the REA LDB to be available on a standalone ship-based platform. This meant we wanted the ability to move the database as a single unit to the at-sea environment.

However, this requirement can be met without importing all data into the LDB. The external data sources could be managed as a remote library (e.g., on DVDs, or remote servers) with an interface built between the REA LDB and the external sources.

It is this type of design that will create the REA PDB as a component of the REAS. The design presented in later sections will utilize a categorization of input data. Those data collected under the responsibility of DRDC Atlantic will be contained within the REA PDB. Those data which represent static products and used by models accessing the PDB, can also be stored within the PDB. The largest design difference occurs for those data that are large volume or frequently updated products from external sources. These data sets will utilize the PDB in terms of it being a catalogue of the external data sets. As well, the catalogue will control the software applications that are used to interface between the data sets and the visualization software. These differences will provide the PDB with the scalability needed for such a system.

4 Standardizing on an oceanographic data model

Research communities have been advocating the benefits of data and information sharing for many decades. The standardization of procedures to deal with oceanographic data collection and processing has a long history, dating back to the early oceanographic programs of the International Council for the Exploration of the Sea (ICES (2009)). As well, procedures that enhance this sharing are being realized. In particular, the use of standards within research fields is allowing researchers to share data and applications among organizations.

In the marine community, there are remarkably few standards or specifications being used for data models. Here, we adopt the Marine Metadata Interoperability project's definitions of standards and specifications as described by Stocks, Graybeal, *et al.* (2009). Although most marine organizations have databases that contain their collected data, few organizations have published data models for those databases. Admittedly, documenting the decisions made during the design process is a tedious activity and thus is often omitted from the formal process.

However, one emerging specification for marine related database design is the Arc Marine model (Wright, *et al.* (2007). Arc Marine was a community-based development that focused on implementation in the ESRI (2009) GIS products. The development community consisted of physical and geological oceanographers, and marine biologists.

The goal of Arc Marine was the specification of a generic framework data model that could be implemented in the ESRI products. However, many of the concepts and design techniques applied in the framework are at a vendor independent level. As well, Wright, *et al.* (2007) describes many case studies involving Arc Marine implementations that can be drawn upon and which demonstrate the power of this evolving design framework.

Although there are a multitude of GIS resources which would assist in handling specific types of data, Arc Marine provides a framework for using disparate oceanographic data types in a cohesive manner. As a framework, Arc Marine is not a single data model but rather a framework for developing a specific and application-oriented data model. Arc Marine provides the very basic tables from which specific tables can be constructed. As well, Arc Marine defines a formal thought process to be used in defining these additional tables.

4.1 Database design practices and the ESRI geodatabase

The construction of a logical data model can be understood in terms of the rules which govern the construction of the specific database implementation. Similar to designing a building (i.e., architectural design), data modelling strives to combine user functional requirements, the business rules that define how data are collected and used, and design rules.

The three aspects can be easily understood in terms of a building. The user requirements are the needs that should be met by those individuals using the building. In other words, the inhabitants have certain functions which must be met by the building; these functions represent the user

requirements. The business rules refer to those rules that the inhabitants have for the use of their building. The business rules are often different, depending on the inhabitants, even for similar buildings. Finally, the design rules are imposed by the construction material or local building authorities. These rules should be followed – but are not always followed. Some rules may be conveniently overlooked or avoided.

For environmental data collection, a business rule can also be understood as a condition which is mandatory for a particular business activity. In the collection of oceanographic data, one business rule may be that for every XBT profile collected, there must be accompanying time and position information. Another example of a local business rule is that each allotment of ship time will result in the assignment of a cruise number or cruise character string (e.g., Q304).

4.1.1 SQL and an Arc Marine geodatabase

The rules for database construction originated in the Digital Equipment Corporation (1992) Structured Query Language (SQL) specification. The specification was formally released in 1992, and is informally known as SQL92. The specification was also issued as an International Organization for Standardization (ISO) standard 9075:1992. The specification covered all aspects of database functionality. Few realize that SQL is a rich and functional language - much more than simple SELECT or INSERT statements. The SQL standard specifies all aspects of the SQL language, including defining statements for database tables, table columns, domains, primary keys, all forms of referential integrity, constraints, procedures, triggers, etc. The most recent edition of the SQL standard was created in 2003, with the total standard now consisting of 14 individual parts. The most recent addition details the eXtensible Markup Language (XML).

For this data modelling exercise, it is instructive to understand how the Arc Marine ESRI geodatabase is constructed relative to the rules set in the SQL standard. The ESRI geodatabase is a general concept, with the Arc Marine data model as described by Oregon State University (2008) being a specification built on the principles of the geodatabase. Arc Marine was developed specifically for use with the ESRI geospatial products, and so Arc Marine utilizes functionality and design decisions that are directly linked to the ESRI geodatabase. Understanding the structure of an ESRI geodatabase is particularly important for the data modelling conducted in this effort.

The ESRI geodatabase is a database built in one of two DBMSs: Microsoft Access or Oracle. Each of these DBMSs adheres to the SQL standard. However, the subtlety is that the geodatabase is defined within the DBMS without using much of the functionality provided in the SQL standard.

This is best described through a specific example. Consider the SQL creation of relationships within the geodatabase concept. Relationships are created within a database to enforce referential integrity. Referential integrity is the concept of enforced limited content between rows from two tables. The SQL specifies the rules for constructing relationships between tables in a database. Foreign key relationships can be formed between tables, if certain conditions apply. These conditions are related to the primary keys of the two tables.

The SQL standard indicates that foreign key relationships can only be created by using primary keys of the referenced or parent table (see 11.8 in Digital Equipment Corporation (1992)). This means a relationship may exist between a primary key and a non primary key or between primary key and primary key (see Figure 1). Thus, if the relationship shown in Figure 1 (top-panel) is formed between Table1 and Table2, then the primary key in Table1 (denoted *Field1*) must be placed in the non primary key in Table2. A second form of relationship is shown in the lower panel, where the primary key of Table3 must be placed in a primary key field of Table4. Note that the SQL92 specification does not permit a foreign key relationship between a non primary key in Table1 (indicated as *Field2*) to any other table.

As noted, the development of Arc Marine was specifically for ESRI products. In the literature that describes Arc Marine and applications of Arc Marine, foreign key relationships that reference non primary fields in the referenced table exist. Such relationships do not comply with SQL; so how is this possible in a typical DBMS and why is it being used?

Such relationships are created within a geodatabase through the use of tables which are unique to the geodatabase. The ESRI geodatabase uses a set of tables within the DBMS to store all relationships. These tables are called relationship class tables. Effectively, the relationship class tables store any relationship that the user wishes to create. These relationships are not restricted by the SQL standard. Thus, the methodology does not utilize the primary key, foreign key or relationship capabilities that are inherent within a DBMS. Since ESRI uses a specific table to store the relationships between tables, there is no longer the strict requirement to follow the SQL92 specification.

There is an important implication to this subtle change. First, the capabilities of the DBMS are not being fully exploited. A typical DBMS adheres to the rules of SQL and given these rules it provides automated functionality such as integrity checking using foreign keys. The ESRI geodatabse implementation does not utilize the DBMS functionality related to referential integrity.

Second, external applications are forced to perform the rules that are realized in the referential integrity checks. When external applications write to the geodatabase, the applications may use either ESRI connection methods, or open database connectivity (ODBC) connection methods. In either case, if the application is required to perform writes of any form (e.g., updates, inserts) to the geodatabase, then the application must be capable of understanding the ESRI relationship tables. If the application does not use the ESRI relationship tables, then the application will not be aware of any integrity constraints. If the application is unaware of the ESRI relationship tables, the application could manipulate the data within the geodatabase without regard to the relationship tables and potentially introduce integrity errors that would only be detected by software that utilizes the ESRI connection methods.

As noted by Isenor and Lapinski (2007), a standard, such as SQL, introduces constraints to the system. In this case, SQL introduces constraints on the relationships between the database tables and fields. Without these constraints, users have more capabilities to create any type of relationship they consider appropriate. This represents a trade off in capability. Although the user can create any type of relationship, they give up data integrity which is a natural consequence of following the SQL standard.

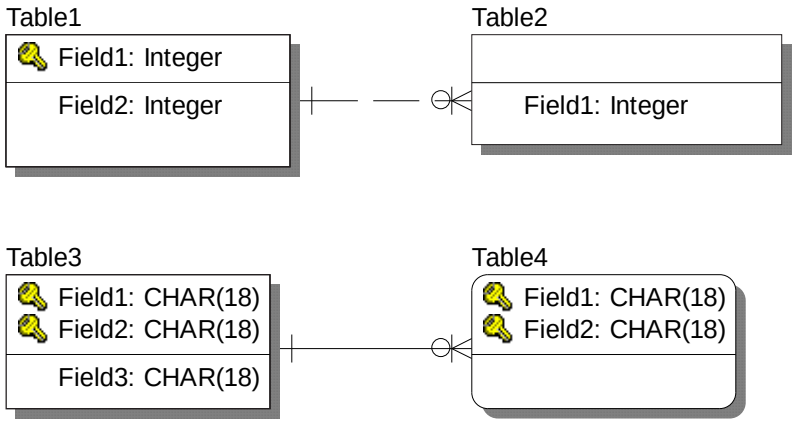

Figure 1: A graphical example of entities, attributes and relationships as shown by the ERwin data modelling software. In the upper panel: a) the primary key Field1 (note the "key" symbol) is related to Field1 in Table2. This is termed a non-identifying relationship. In the lower panel b) the primary key, composed of the composite of Field1 and Field2, is related to the identical fields in the primary key of Table4. This is termed an identifying relationship. Note that Information Engineering (IE) notation shows the panel a) relationship with a dashed line, and panel b) relationship as a solid line. See also section 6.3.1 for a description of the symbols on these relationship lines.

4.1.2 Relationships in Arc Marine as compared to SQL

As noted in the previous section, the Arc Marine method of dealing with data relationships is different from the SQL standard. However, we must acknowledge that complying with the SQL standard introduces certain issues related to the distribution of primary key values over multiple foreign key fields. Again this is a design trade-off that is introduced by the implementation method that we select.

The issue is best described using an example: the *Feature_ID* field. This field contains a numeric identifier that is unique to all features in the database. These features may be points, lines, areas or other geospatial representations. In database terminology, the *Feature_ID* is a surrogate key (Pascal (2000)).

The *Feature_ID* of a point may be contained in a particular table, for example the Instantaneous_Point table. The *Feature_ID* of an area may be contained in a different table, called Feature_Area. Each set of *Feature_ID* values in the two tables are unique in the table; but, the values are also unique across the tables. So, any *Feature_ID* value existing in Instantaneous_Point cannot exist in Feature_Area.

SQL itself provides no means of verifying uniqueness across the two tables. In such a case, a parent table can be introduced to contain all defined *Feature_ID* values (i.e., call this parent table

Feature_Asset). Both child tables (i.e., Instantaneous_Point and Feature_Area) contain *Feature_ID* as a foreign key relationship to Feature_Asset. The relationship guarantees that the *Feature_ID* numeric value in Instantaneous_Point and Feature_Area exists in Feature_Asset. However, the relationship does not control the uniqueness of the numeric across Instantaneous_Point and Feature_Area. In other words, the database structure cannot by itself, guarantee that a numeric *Feature_ID* in Instantaneous_Point is unique when compared to the set of *Feature_ID*s that exist in Feature_Area (the opposite also being true).

To ensure the uniqueness across the Instantaneous_Point and Feature_Area tables, a programmatic check is required. Ideally, this must be done on each insert and modify operation of any of the tables involved in the relationship. This programmatic solution can be costly in terms of computer time.

4.2 Spatial reference systems and frames

One of the issues raised by Deveau (2008) at the end of the Phase II work was related to the common spatial reference identifier (SRID). Specifically, the report recommended the conversion of all positional data to a common SRID. Understanding the SRID and its implications for GIS is important. Here, we explain the concept of the SRID, the spatial reference frame, and finally the implications on the data.

We first issue a word of caution. The terminology used in this field appears to be inconsistent. In some cases, distinctions are drawn between coordinate reference systems (CRS) and coordinate reference frames (Craymer (2006); Seidelmann (1992); Junkins and Garrard (1998)). In other cases the coordinate reference frame is termed a georeferenced coordinate reference system (International Association of Oil and Gas Producers (2006)). In this description, we will follow the practice of the European Petroleum Survey Group (EPSG).

To begin the description, we must first consider a model for the Earth. A model in this context, is a mathematical representation of the Earth. There are many potential models for the Earth, including a spheroid, a rotational ellipsoid or a triaxial ellipsoid. The trade off as we move through these models is the errors in fitting the model to the actual shape of the Earth, versus mathematical complexity. The most often used compromise is the rotational ellipsoid.

A rotational ellipsoid (hereafter referred to simply as ellipsoid) is a mathematically defined shape that is in fact fully defined by specifying only the lengths of the two axes of the ellipsoid (or alternately, one axis and the flattening ratio). Defining the two axes actually defines an ellipse, while the ellipsoid is determined by then rotating the ellipse about its minor axis. However, to use this mathematical shape as a model of the Earth, we must position the ellipsoid relative to the Earth. This involves assigning the centre of the ellipsoid to a known location, such as the centre of mass of the Earth. This method of fitting is referred to as geocentric. As well, the z axis of the ellipsoid can be assigned to be parallel to the Earth's axis of rotation. Finally, the x axis of the ellipsoid is assigned an intersection with the Earth, which is typically the Greenwich meridian. In positioning the ellipsoid relative to the Earth, we create a georeferenced coordinate reference system.

The ellipsoid is typically positioned relative to the Earth in either a local or global sense. In most cases, it is local, in that local geodetic control points (also called datum points) are used to adjust the ellipsoid to best fit the local geopotential surface, known as the geoid. Since the control points are local, the best fit to the ellipsoid only applies locally. A different Earth location using a different set of control points will create a different fit to the ellipsoid. By using a different set of control points, one alters the centre and orientation of the ellipsoid. Note that this alters the positioning of the ellipsoid and not the mathematical definition of the ellipsoid.

By using the same ellipsoid definition we only use one coordinate reference system. The different fits of data points to the ellipsoid result in multiple georeferenced coordinate reference systems (GCRSs).

The use of local geodetic control points to fit to the ellipsoid results in many defined GCRSs. These systems are important for any geospatial activity. The petroleum industry has a particular interest in GCRSs. As a result, the EPSG has taken the task of central authority for management of the GCRS. Part of this management process involved the introduction of the spatial reference identifier (SRID). The SRID is simply a unique numeric identifier for a georeferenced coordinate reference system (GCRS). These identifiers are maintained by EPSG. Note that in 2005 the EPSG was reformed into the Oil & Gas Producers (OGP) Surveying and Positioning Committee.

The CRS is a mathematical model used to define the ellipsoid in a similar way that $y = ax + b$ is a mathematical model used to define a line. The CRS applied to the Earth has changed through time. As well, the georeferencing has also changed through the fitting of different data to the CRS. Using different data sets within the same CRS will result in different GCRS being formed. In the same way, different data points for fitting a line results in different coefficient values for a and b. The definitions of GCRS date back to the early 1800's.

The information that references the ellipsoid to the Earth is collectively called the geodetic datum. The datum includes the information noted above, plus such things as the astronomical coordinates of the ellipsoid. There are also horizontal and vertical datum which are subcategories of geodetic datum. These particular datum serve to position points either in the horizontal or vertical.

The location of a point on the surface of the Earth is expressed in the GCRS, typically in latitude/longitude/height values. These measurements are actually defined based on the point location relative to the ellipsoid as defined by the GCRS. Thus, each change in the GCRS potentially results in the specific point on the Earth having a different numeric position (i.e., a different latitude/longitude/height).

Consider latitude as an example. The latitude measurement used for the Earth has many definitions. Here we consider common latitude (Wikipedia (2009)), which is defined as the angle between the equatorial plane and the extension of a line normal to the surface of the ellipsoid. Note that for an ellipse, the normal from the surface does not necessarily intersect the centre of the ellipsoid. Intersection with the centre would only occur if the ellipsoid were a sphere. The angle formed by the normal to the ellipsoid and the equatorial plane is the common latitude.

As the GCRS changes, there is the potential for the ellipsoid to change. In turn, this can change the latitude measurement for a specific point on the Earth. Alternately stated, the latitude of a point on the Earth can change due to a change in the GCRS. Changes in the GCRS can in fact

result in the same Earth point being given different positional values in all three coordinates: latitude, longitude and height.

The OGP Geodesy Subcommittee (2008) has constructed a database of GCRSs. This database contains a listing of past and present ellipsoids and GCRSs. The OGP lists 48 ellipsoids in its data set. One of these ellipsoids is known as the Clarke 1866 ellipsoid. The OGP database can be searched to determine that the Clarke 1866 ellipsoid was used to define 245 GCRSs, these being both global and local.

Currently, the GCRS in common use is the World Geodetic System of 1984 (WGS84). This is the GCRS being used by the global positioning system (GPS) satellites. Previous to this, Canada used the GCRS known as the North American Datum of 1983 (NAD83). Craymer (2006) describes two realizations of this CRS (note the omission of the "G") - NAD83 and NAD83(CSRS96). These two systems were both based on the same ellipsoid, but the latter used an updated control point data set in its definition.

NAD83 was introduced in 1986. Previous to this, Canada used the North American Datum of 1927 (NAD27). This ellipsoid was specifically positioned to best represent the North American continent.

To illustrate the scope of the GCRS issue, consider that there are 197 GCRS definitions for NAD27 in the OGP definition database. All of these GCRSs use the same ellipsoid; the Clarke 1866 ellipsoid. The various definitions have been fitted using local data. Of the 197 NAD27 GCRSs, 40 pertain to Canada. As an example, NAD27(76) (SRID 4608) was one of 15 GCRSs used in Ontario. In Quebec, a total of 21 GCRSs for NAD27 exist in the OGP database.

4.2.1 Implications to DRDC Atlantic data

A GIS application places features (e.g., points, lines, areas) relative to the Earth and relative to one another. The GCRS used by the data in the GIS is critical to the overall system. GIS systems must know the GCRS of all positional data within the system. Without knowledge of the GCRS, the GIS cannot properly place the data on a display. Alternately stated, knowing the latitude and longitude of a data value does not uniquely position that value on the Earth or on a display. The GIS must also know the GCRS from which the latitude and longitude were measured. Only then can the value be uniquely positioned.

As well, the multiple sources of data must either use the same GCRS, or have available a transformation between the GCRS used for the data and the GCRS used on the display. Transformations between GRCSs are available from the OGP in the form of transformation files which are dependent on the position, or in the simpler cases as linear transformation values.

The data collected by DRDC Atlantic does not explicitly indicate the GCRS which was used during the collection. Thus, in most cases we simply have to make an educated guess as to which GCRS was in use by the systems reporting the positional data. The earliest data found within the REA LDB is from 1977. Thus, it is quite possible that observational data within REA LDB has been collected using NAD27, NAD83, and WGS84. In such a case, the differences introduced by

these GCRSs should be quantified in order to understand the importance of changes to the reference system.

The difference between the reference frames is not constant through space. To quantify the differences introduced by the different GCRS, we need to consider the spatial location of the data. Using XBT data as an example, the REA LDB contains profiles over the Scotian Shelf, Grand Banks of Newfoundland, near the United Kingdom, the Mediterranean Sea, and near the Bahamas.

Transformation software, developed by Geomatics Canada, is available to transform NAD27 positions to NAD83 positions. However, the software was developed for continental North America and does not cover the entire North Atlantic ocean. Nevertheless, the software will provide an estimate of the errors we may expect within the REA LDB positional data.

We consider three positions as indicated in Table 1. The software indicates that the transformation between NAD27 and NAD83 introduces positional differences of 27-131m. If the NAD27 or NAD83 spatial reference frame was also used during data collection near the UK coast or the Mediterranean Sea, the errors in positioning would be larger – unfortunately we do not know how large. The transformation software limits the spatial domain and does not permit transformations at those locations.

Fortunately, all XBT drops east of 15ºW were collected in or after 1993. Thus, it is unlikely those points used NAD27 but they may have utilized NAD83 or possibly NAD83(CSRS96). Differences between NAD83 and NAD83(CSRS96) over Canada are less than 2m while differences between NAD83(CSRS96) and WGS84 are about 1m in the horizontal throughout Canada (Craymer (2006)). As noted above, if the NAD83 were used at such a distance from the North American continent, the differences would be larger than the stated differences.

The differences stated here need to be placed in context relative to the applications for the data. If we were dealing with near-shore navigational data, such differences would be important. For example, the narrows in Halifax Harbour is approximately 380m wide. In a GIS system displaying the narrows, a 100m error introduced by using the incorrect reference frame could easily place a known water asset (e.g., a ship) on dry land.

However, our ocean data were collected on the shelf and deep ocean. The research ship CFAV QUEST, which was used for the more recent data collections, is 76m in length. The GPS receiver on the Quest is located over the bridge, while the XBT drop point (i.e., using XBT data as an example) is at the stern. We can assume a difference of about 50m between receiver and drop point. Note that this positional error is considered unimportant for the research activities. Also note that when the ship is steaming forward, and in the case of the deployment taking place at the aft of the ship, the displacement between deployment and GPS receiver increases due to the time it takes to notify the bridge of the drop.

Given the position accuracy of the XBT drop point compared to the ship GPS receiver, the implications of steaming while deploying, and the usage requirements of the data, we consider the GCRS conversion difference as unimportant for our studies.

Table 1: Three typical locations are used to illustrate positional errors introduced by GCRS. The transformation between NAD 27 and NAD83 is shown to introduce errors of between 27 and 131 metres.

Location	Latitude (ºN)	Longitude (ºW)	NAD27 / NAD83 Difference (m)
Emerald Basin	44	63	56
Eastern XBT Extent	41	45	131
Southern Extent of Software	40	76	27

5 GIS basics

Geographic Information System (GIS) refers to the hardware and software that specializes in the storage, display, analysis and manipulation of geospatial data. The GIS had its beginnings in computerized mapping but was soon extended to the more general display of all types of georeferenced data. An overview of GIS is provided by Buckley (2009).

5.1 PostGIS

PostGIS (2009) is a GIS add-on specifically developed for the PostgreSQL database. PostGIS allows the creation of geographic objects in the PostgreSQL environment. Effectively, this means the PostgreSQL database then supports the geometry encodings for a GIS spatially enabled database. In turn, this means the PostgreSQL database can then be used as a backend to a GIS. PostGIS was developed by Refractions Research. It is open source and is released under the GNU (2007) General Public License.

The PostGIS manual (see Ramsey) identifies numerous data encodings as supported by PostGIS. These are:

- POINT – individual points are contained in the geometry
- LINESTRING – a set of points that are connected together to represent a single line are contained in a single geometry
- POLYGON – a set of points that are connected together to represent a polygon are contained in a single geometry
- MULTIPOINT – multiple independent points are contained in a single geometry.
- MULTILINESTRING – multiple independent lines are contained in a single geometry
- MULTIPOLYGON – multiple independent polygons are contained in the geometry
- GEOMETRYCOLLECTION – a collection of different types (e.g., POINT, LINESTRING, etc.) are contained in a single geometry

The simplest of these is the POINT type. The encoding of a point means we use the latitude, longitude and possibly a depth value to compute the encoded geometry value. An example of the geometry is shown in Table 2. The table illustrates that a simple point in latitude, longitude, depth space of 43,-63, 20 becomes a complex encoded value.

In a GIS database, the common latitude, longitude, and depth values would be represented as a geometry encoded value as shown in Table 2. Thus, the actual latitude, longitude and depth are not directly available as numerical values. The benefit of the encodings is in spatial query optimization. Special algorithms in the GIS environment improve search speed of the encoded

values as compared to storing numerical values of latitude and longitude. Also, additional tools in the GIS environment allow specialized querying for such conditions as the intersection between objects, overlaps between objects, parallel objects, etc.

Table 2: The geometry point type is shown. The GIS uses the latitude, longitude and depth values with the spatial reference frame (in this case SRID=4269) to encode a special point geometry value as shown in Point Geometry column. Having the ability to create and utilize these encodings means the database is spatial enabled.

Lat. (ºN)	Long. (ºW)	Depth	Point Geometry
43	63	20.0	01010000A0AD10000000000000000804FC0000000000080454000000000000003440
43	63	NULL	0101000020AD10000000000000000804FC00000000000804540

5.2 uDig

To utilize a GIS enabled database, we need to have GIS software that understands the geometry encodings of the GIS database. This complicates matters because encodings are specialized to particular software producers. Thus, an encoding produced by PostGIS will not likely be understood by a different software package (e.g., ESRI Arc Map).

For PostGIS, the User-friendly Desktop Internet GIS (uDig) is available. uDig was developed by Refractions Research (2009) and is offered using the GNU (2009) Lesser General Public License (LGPL). uDig was developed with the financial support of GeoConnections (2009) Canada. For this brief introduction, uDig version 1.1.1 was used.

The uDig display is shown in Figure 2. In typical GIS fashion, the left panel shows available maps (upper portion) and the layers for the current map (lower portion). The top-right panel shows the current map. For this particular example, the Nova Scotia region is shown.

Within the map display are the map layers. Each data set can have an individual map layer. In Figure 2, the green shaded map areas are the CF operation areas (know as Op Areas, see Department of National Defence (1992)). These Op Areas are contained in one layer of the GIS. The black squares represent a second layer, and in particular are the locations of bedrock outcropping. A third GIS layer is represented by the orange contours of Scotian Shelf surficial geology as provided by Gareau (2005).

Since uDig is open source, plug-ins have been developed specifically for enhancement to the core uDig functionality. One plug-in provides spatial operators which compute the intersection of

layered data (other functionality is also provided by the spatial operators). These operators allow the user to query the intersection of the layers. In this example, the query can act on the bedrock outcrop layer and determine the intersection with the surficial geology layer. This effectively identifies the surficial geology type at all the points identified as a bedrock outcrop. The results of the query are shown in the bottom panel in Figure 2.

uDig can access the PostGIS enabled database directly, and understands the encodings of the GIS database. uDig is also capable of understanding view tables from the database. Finally, uDig can import ESRI shapefiles.

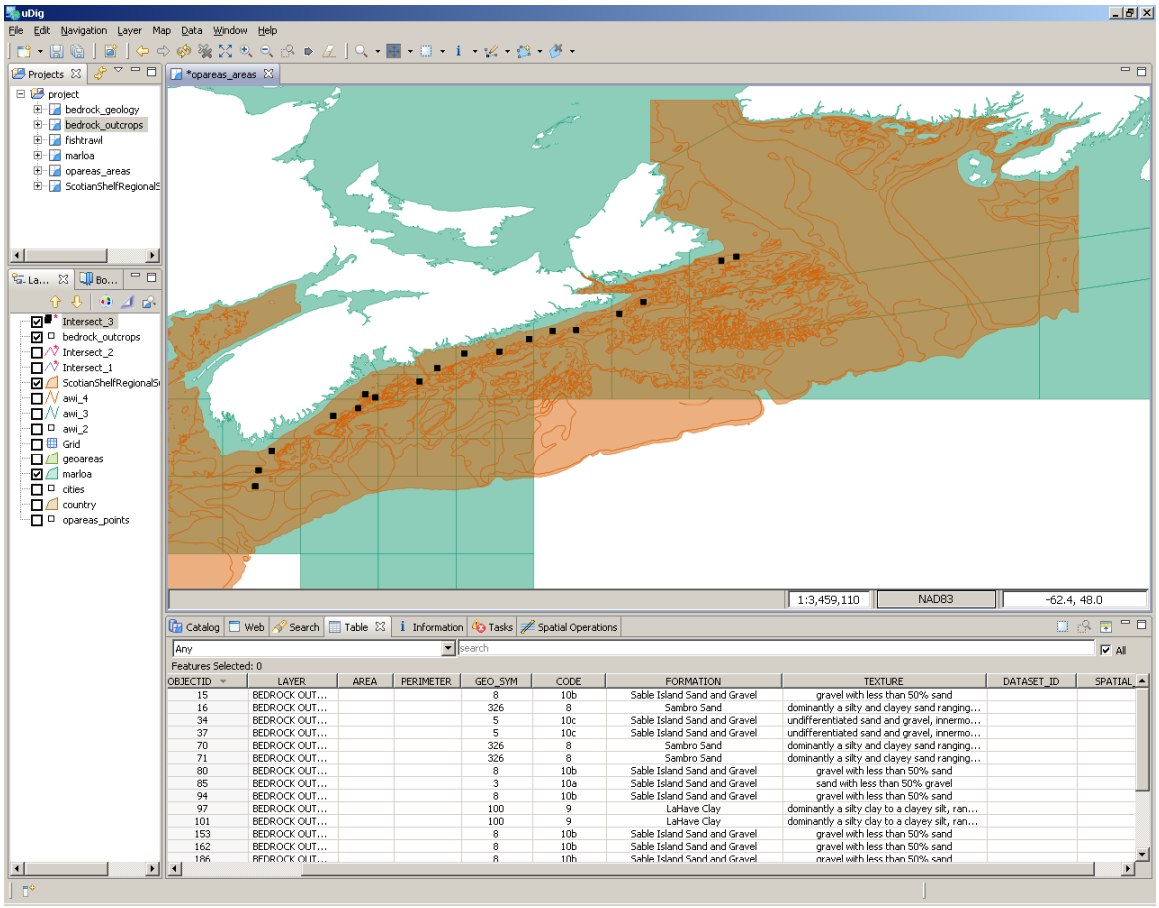

Figure 2: The uDig interface. The left panel displays available layers for the current map. The top panel displays the map. The Nova Scotia region is shown. The green shaded areas are the CF operation areas (know as Op Areas). These Op Areas are in one layer of the GIS. The black squares represent the locations of bedrock outcropping and are in a second layer. A third GIS layer is represented by the orange contours of Scotian Shelf surficial geology. uDig spatial operators allow one to query the bedrock outcrop layer with the surficial geology layer, identifying the intersection of the two layers. These data were obtained from shapefiles on the Geoclutter CD-ROM (see Gareau (2005)).

6 Data modelling for the REA production database

6.1 Analysis of the existing REA LDB

The data modelling exercise began by first developing a list of all tables and field names in the REA LDB version 3 beta. The list was constructed using the Computer Associates ERwin Data Modeler software version 4.1.2522, performing a reverse engineering function on the PostgreSQL database using a Postgre ODBC connection. This procedure identified 100 tables with 737 columns. This is after the removal of two extraneous tables that have been created by users of the REA LDB.

The list of table and field names was then moved to a Microsoft Excel spreadsheet. This spreadsheet was then used as a primary documentation tool, to document the meaning of all fields in the existing LDB. This documentation effort is required to fully understand the content of the fields. This level of documentation was not completed in the work reported by Deveau (2008) but is critical to allow us to determine if the content of a particular field should be included in the port to the PDB.

Table 3: An example of one table and associated field names from the existing REA LDB. The process followed in this work required that each field name be investigated to determine an appropriate field definition. This allows us to understand the field content and thereby decide whether or not the content should be included in the PDB.

Table Name	Field Name	Field Type	Field Definition
opareas_areas	*id*	integer	Primary key. Sequential counter.
	id_oparea	integer	The primary key "id" counter from the opareas table.
	notes	text	Only 3 records present were notes in the coordinates field from the DND publication that defined the areas.
	charts	text	The charts that the_geom polygon applies to.
	the_geom	geometry	The geometry as defined by POSTGIS system.
	bounds	boolean	Contains text values "t" or "f"; indicating true or false. This likely indicates whether or not there is complete overlap of the chart and the op area.

An example of the required documentation is shown in Table 3. The LDB table opareas_areas is used in the REA LDB to describe the physical areas associated with CF operation areas. Each operation area record has a unique *id* for the record. As well, each record has an identifier, *id_oparea*, which identifies the operation area. The *notes* field is a free text field for any comments that the data administrator may have. The *charts* field indicates those charts which overlap in any extent with the operation area. This is an artefact of the paper-based system used to describe the operation areas. In a GIS environment, such a field should be replaced with the spatial extents of the charts, thus allowing any operator to overlay charts and operation areas in their display. The last field, *bounds*, is a boolean which indicates if the operation area is fully described by the specific chart. Again, this is an artefact of a paper-based system.

The entire list of tables in the existing REA LDB is provided in Table 4. From the list of 100 tables, we can identify 9 tables as not required for the PDB; those beginning with _ or z1, since these are temporary tables using during the initial data import. As well, the 8 tables from the British Oceanographic Data Centre (BODC) were never intended to be part of the LDB and thus these can be ignored (the information pertaining to the BODC parameter codes was intended to be included, but not the tables themselves). Tables categorized as "data import administration" and "other" can also be ignored, bringing the total to 24 tables; thus leaving 76 tables that contain data that must be accounted for in the PDB.

Table 4: The existing tables as grouped according to categories of convenience. Of the 100 tables listed, only 76 need be considered in the data port to the production LDB.

Assigned Category	Number of Tables	Table Name
Begins with _ (underscore)	3	_atable
		_temp_nadas_lines_decoded_parts
		_temp_view_xbt_file_meta_data
Begins with z1 (indicates a load table)	6	z1_nadas_filenames
		z1_nadas_lines
		z1_swdb_filenames
		z1_swdb_lines
		z1_xbt_filenames
		z1_xbt_lines
BODC related	8	bodc_biota_comp_model
		bodc_category
		bodc_category_link
		bodc_chem_model

		bodc_itis_map bodc_parameter bodc_parameter_group bodc_units
East Coast Ambient Noise	4	ecs_flights ecs_sites ecs_wind_obs ecs_wind_obs_vs_model
Bathymetry related	6	atlbathy bathy_filenames bathy_lines etopo2 etopo5 wh_depth gom15dd
Geoscience Canada related	25	gc_airgun_profile_sections gc_bedforms_basinatlas gc_bedforms_frm_sidescan gc_bedrock_geology gc_bedrock_outcrops gc_bouyancy_line_moraines gc_drift_outcrops gc_ed_nav_of1427 gc_epicentres gc_faults gc_fishtrawl gc_groundfish gc_gsca_ship_tracks gc_iceberg_furrows gc_infilled_channels

		gc_isopach_contour gc_isopach_thickness gc_nearshore_bedrock gc_nongeoclutter_surveysitespoly gc_pockmarks gc_rib_moraines gc_sand_ridges_chs gc_scotian_shelf_regional_surfical_geology gc_seabed_texture_frm_sidescan gc_till_tongues
Shallow Water DB Related	4	swdb_geocircles swdb_geopoints swdb_index swdb_tl_files
NADAS related	3	nadas_codes nadas_file_meta_data nadas_observations
Bellhop related	3	bellhop_data bellhop_plots bellhop_q
GIS related	7	geoareas geocircles geolines geometry_columns geopoints spatial_ref_sys
Operation Areas	3	opareas opareas_areas

		opareas_points
Temperature/Salinity grid	3	tsd_geopoints tsd_salinity tsd_temperature
XBT	2	xbt_file_meta_data xbt_profiles
Data import administration	3	directory_specifications filenames filetypes
Scotian shelf sediment	2	sediment_ss_position sedthick
User administration	3	authorization_table logins userqueries
Data set administration	4	cruises scientists scientists_vs_filename ships
DRDC Units and code types	3	codesources drdc_units drdctypes
Other GIS input	2	cities country
ISO related	1	isocodes

MARLOA related	1	marloa
Other	4	data_columns testme type_xref_column_name type_xref_table_name

6.2 Components of the conceptual model

A conceptual model is presented (Figure 3) to provide the reader with overall knowledge as to what the data model encompasses. The model provides storage for data of two primary categories: point and mesh. The categories for lines and areas are also present in the model, but these categories support the point and mesh data. Surrounding the point and mesh categories are the metadata associated with the management of these categories. These management functions require information on particular cruises, instrumentation, parameters and data packages. There are also metadata associated with specifics of a cruise. Finally, the management of the processing applied to the data is contained in a specific lineage section of the model.

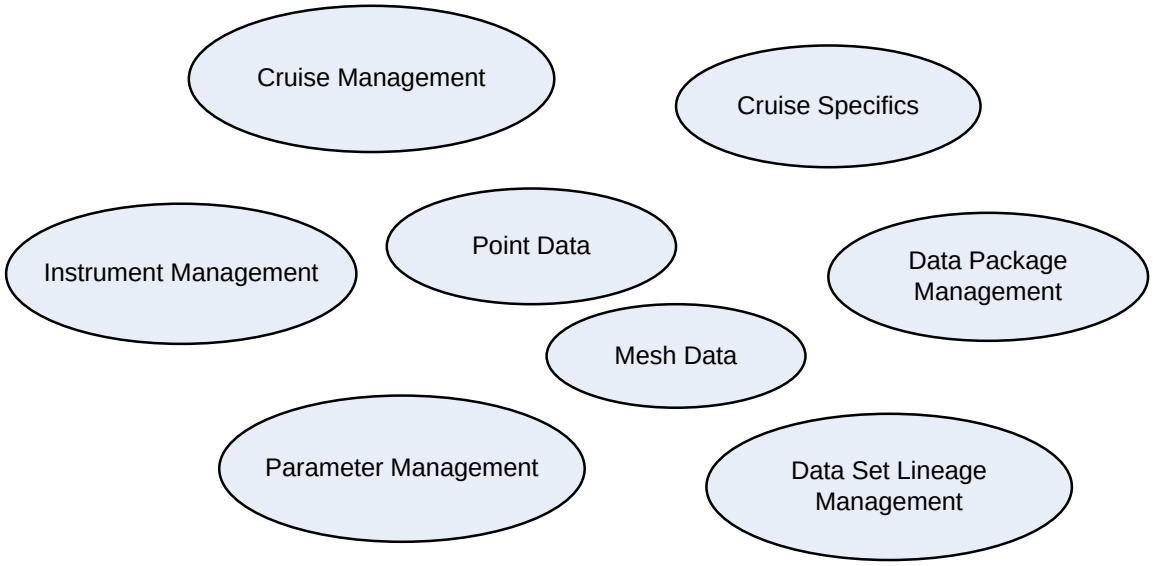

Figure 3: The components of the conceptual model. Many components are related to the management of the data within the PDB. At the centre of the conceptual model is the point and mesh data.

6.3 Data modelling for the production database

For this effort, a spiral approach was used to develop the data model. The initial data model considered a single specific data type, with the data model designed to address that data type. After each development, another data type was examined with appropriate data structures added to the data model. Existing data structures were often refined in this process. As more data types were considered, the data model spiralled to the design described in this report.

The following sections examine each data set considered during construction of the data model. Any one data set may be considered as one or more individual data types. Throughout this discussion, the data already existing in the LDB were mapping to new storage locations in the PDB. This mapping is an extremely detail oriented task and as such, is both time consuming and error prone. The details of the mapping are available in Annex A. Annex A can be used in conjunction with the sections below, to understand how the individual data values as stored in the LDB are mapped to new storage locations in the PDB. A complete list of PDB table names and comments is provided in Annex B.

6.3.1 Vertical Profile data

The data modelling for the REA PDB was also conducted in Computer Associates ERwin version 4.1.2522. The modelling consists of constructing entity and attributes that ultimately become database tables and fields that will store the data residing in the REA LDB (version 3b) tables identified above (i.e., the 76 tables). The IE notation is used in this modelling. A review of IE notation is provided in Figure 4.

To begin the process, it was decided to first deal with a data set that was well known by the authors – general vertical profile data. Specifically, the LDB contained vertical temperature profiles from XBT instruments. Since the vertical profile data was the first geometric type considered, we also had to create those foundation tables required for the administration of the data within the database.

In total, 21 tables were created in this initial construction. This initial construction was larger than most because the foundation or management tables were also required. The table names and associated table comments are provided in Table 5.

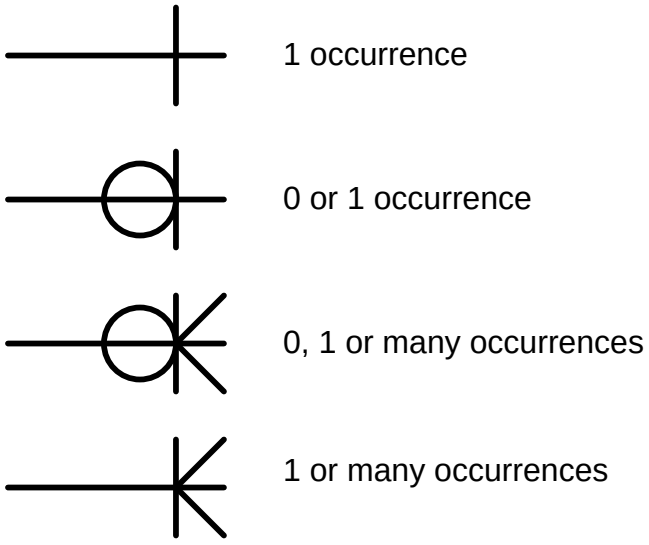

Figure 4: The crows-foot notation used in the data modelling. Formally, this is known as Information Engineering (IE) notation. These lines, when attached to an entity, indicate occurrences of common values that are permitted between entities.

Table 5: The initial set of tables required for vertical profile data (e.g., XBT data).

Table Name	Table Comment/Description
Cruise	The main table to declare a cruise as a data collection activity.
Cruise_Notes	Cruise_Notes contains any notes to be associated with the cruise.
Data	The Data table was initially split into Measured_Data and Computed_Data. This complicates the model because it provides a split of values across two tables. We have revised this numerous times, and now consider one table to be a valid solution. Computed data will have "COMPUTED" as the device (which is linked via Device_ID). In the case of sound speed data, there may be a valid device (e.g., XSV) or COMPUTED device. We have also added an UNKNOWN device for those data that we are unsure of origin. Finally, we added Replicate_ID as a counter for replicated measurements from the same device. Note that device does not indicate a unique device (i.e., serial number) , but rather a model of device. Devices could be added for uniqueness, but this was not the initial intent. Also, COMPUTED could be separated into various computation methods if so desired.
Data_Packages	This is the master table that identifies specific data assets or resources. The Asset_ID has RoleName of Cruise_ID in other tables. This type of structure allows the incorporation of non-cruise data sets into the data model. For example, if model output is included it would not technically be related to a cruise. In this case, the Asset_ID would increment for the model output and likely be stored in non-cruise related tables.
Device_Class_Detail	This table provides information on the details of a specific class of device. This is NOT information specific to a particular device. As an example, the coefficients that should be used to process an XBT cast for a specific XBT type (e.g., T5, T7) would be noted here. Then, the actual values used in the processing are noted in the Value field.
Feature_Asset	Table which lists all Feature_ID values that apply to a specific Asset_ID. This list of Feature_ID values is then subdivided into one of many possible tables in the lower structure of the data model.
Instantaneous_Point	An Arc Marine table that contains information on single space-time measurements or computed values.

Measurement_Location	The spatial location of the measurement. The Arc Marine model has this table named "Measurement". The name in this implementation was changed to more clearly identify the content. Note that a unique Feature_ID defines all member records of a single feature. The Measurement_Location table contains the X,Y,Z points even though for profile data, this violates 3rd normal form (there are no nonkey attributes which determine other nonkey attributes - in the profile case, the Feature_ID and Feature_Class determine the X and Y location values). However, it is considered better from a GIS perspective to keep the coordinate values together. Keeping them together allows the GIS tools to access and use the coordinates in a single geometry, thus allowing the slicing of the data into horizontal or vertical regions.
Measuring_Device	A description of all devices that could be used in the data collection activity. For measured data, it is typical to have a device description. However, for historic data the device related metadata may not exist. In this case, UNKNOWN is used as the device.
Parameter	Contains all the parameters used in the Data table.
Position_Code	Contains the position flags that pertain to the latitude/longitude positions.
Profile_Notes	Any notes that were collected and pertain to the particular profile.
Quality_Flag	Contains the quality flags for the data values. All quality flags are listed in this table.
Scientist	The names of all scientists that have been associated with data collection activities.
Scientist_On_Cruise	The names of the scientists that were involved in a particular cruise.
Series	The Series table is simply a means to collect together a group of Feature_IDs. The uniqueness of the Series_ID is maintained in the Series table. The table allows multiple Feature_IDs to be assigned to a single Series_ID. This allows the set of Features (i.e., set of Feature_IDs) to be grouped together to represent a line, or area.
Ship	The names of all ships associated with any data collection activity.
Ships_On_Cruise	The names of ships that took part in particular cruises.
Track	A track is considered a line along which the ship moves. There may, or may not, be data collected during the transit of a ship. Likewise, there may, or may not, be data collected along a track.

Vehicle	This table describes the vehicle on which a measuring device is attached. A single vehicle can contain more than one measuring device. For example, a towed vehicle could have separate devices measuring temperature and pressure; a Remotely Operated Vehicle could carry an assortment of measuring devices; a marine mammal could be tagged and thus carry multiple measuring devices.
XBT	Contains attributes that are particular to the XBTProfile feature class. These attributes deal with assumed data values and processing details.

Certain tables are important with respect to general vertical profiles. These tables are shown in Figure 5 (due to space issues, not all tables described in Table 5 are shown in Figure 5). The upper table, Data_Packages will contain a unique identifier for each package of data added to the PDB. This table is linked via a relationship to Cruise. The *Data_Package_ID* present in Data_Packages is renamed to *Cruise_ID* in the Cruise table (in other words, the relationship links *Data_Package_ID* and *Cruise_ID*).

The specific cruise record can have associated tracks. A Track record is simply a description of one leg of a ship track with the leg being a single or multi-segment line. Multiple tracks can make up the entire cruise. Along any track, a Vehicle may be used to collect data. A Vehicle can carry many devices (i.e., instruments or sensors) to measure a data quantity. These devices are listed in Measuring_Device.

During the cruise, the instrumentation may make point measurements. These points are described in Instantaneous_Point. Instantaneous_Point is a general table for describing points where measurements were taken. However, some metadata will only be associated with specific profiling techniques. For XBT data, these metadata are in table XBT.

Each point in Instantaneous_Point is assigned a *Feature_ID*. This identifier is used in the relationship to Measurement_Location, where the x,y,z values for all measurements are stored. The x,y,z values are stored in the GIS geometry field, *Geom*. The z component of the geometry is optional. Each set of measurements at a single x,y,z are assigned a *Measurement_ID*. The relationship to the table Data establishes a link between *Measurement_ID* in Measurement_Location and the Data *Measurement_ID*.

The actual data values are stored in *Data_Value* within Data. The type of data is identified using *Parameter_ID*. Each parameter is described using the records in Parameter. Quality flags may be assigned to individual data values using the quality flags stored the Quality_Flag table. A complete list of field names and comments is provided in Annex C. As well, Annex D provides a full list of validation codes used for content in various attributes in the model. One attribute pertaining to this section is *Feature_Code* in Instantaneous_Point.

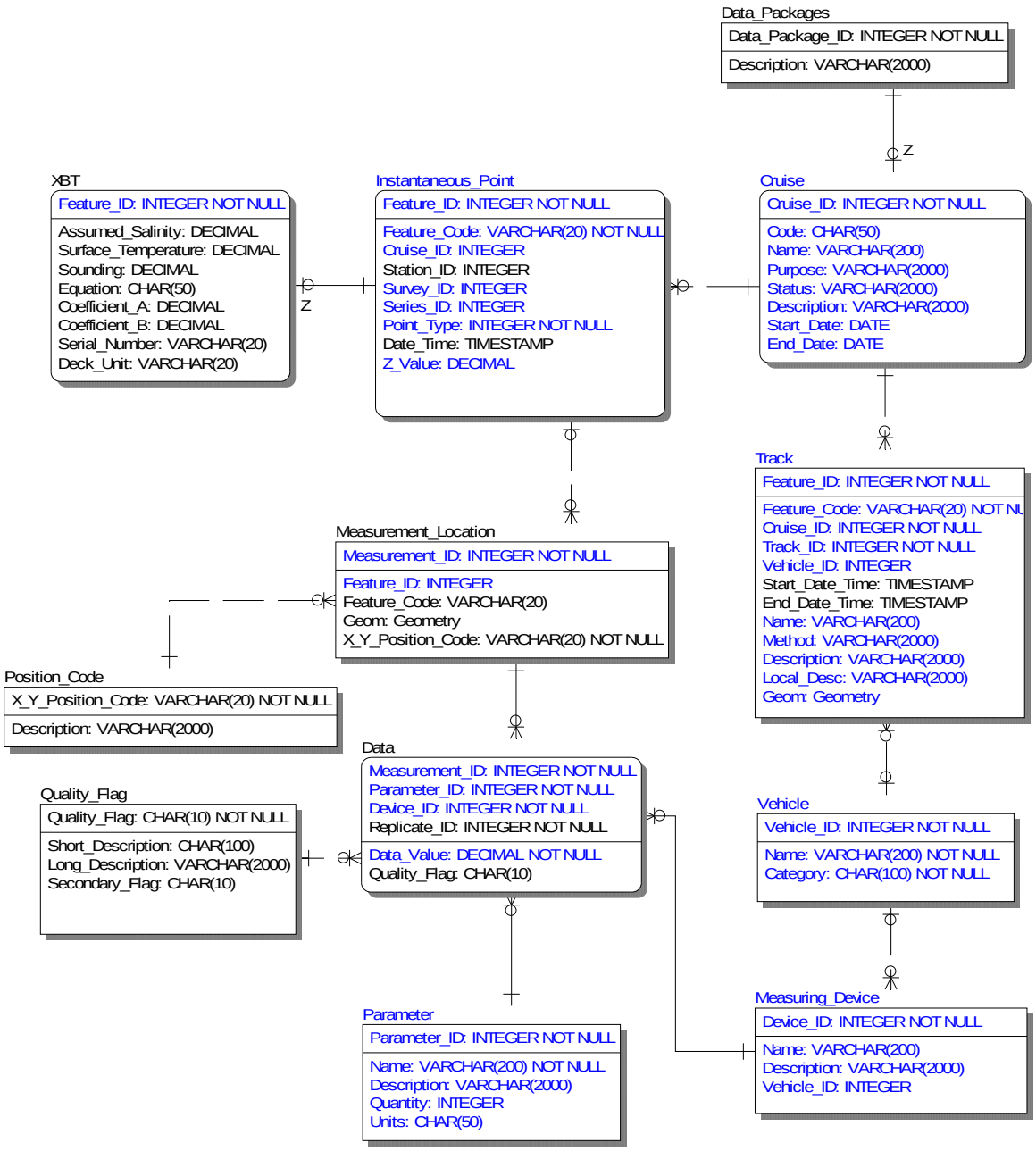

Figure 5: The initial tables used for vertical profile data. The blue text indicates names that are based on the Arc Marine model. Black text indicates extensions to the Arc Marine model for the purpose of DRDC Atlantic data collection activities or business rules. An uppercase Z indicates a zero or one relationship.

6.3.2 Typical shapes of profiles

Some historic data reports have compiled vertical profile data into typical shapes for areas of the ocean. The reports date from the 1950s and present a compilation of temperature data in the form of typical profile shapes for regions of the ocean. The shapes are descriptive only – they have no numeric values for either depth or temperature.

There is no actual requirement to store these typical profile shapes within the database. In fact, the storage of such shapes represents a slight complication, since the shapes do not have associated numeric values. We nevertheless recognize the potential importance of the historic compilation and if possible would like to permit the storage of typical shapes within REA PDB. To allow the storage, we introduce the tables Feature_Area and Area_Characteristic.

The Feature_Area table provides the ability to uniquely identify an area via the *Feature_ID* numeric (Figure 6). Feature_Area defines and names the specific area. As well, the general category or class of the area is described using the *Feature_Class* field. The specific name of the area is specified in the *Feature_Code* field. This allows us to categorize areas into groups, as well as name specific areas within the indicated group. The actual bounds of the area are specified in a polygon contained in the geometry field named *Geom*.

Area_Characteristic table is used to describe a characteristic of the area, which itself was defined in Feature_Area. An environmental variable described using a typical shape profile is considered a characteristic of the area. The *Name* field within Area_Characteristic is used to assign a name to the characteristic. The actual typical shape must be stored as a graphic since no numeric values exist to define the shape. A link to the graphic can be provided within the *Value* field.

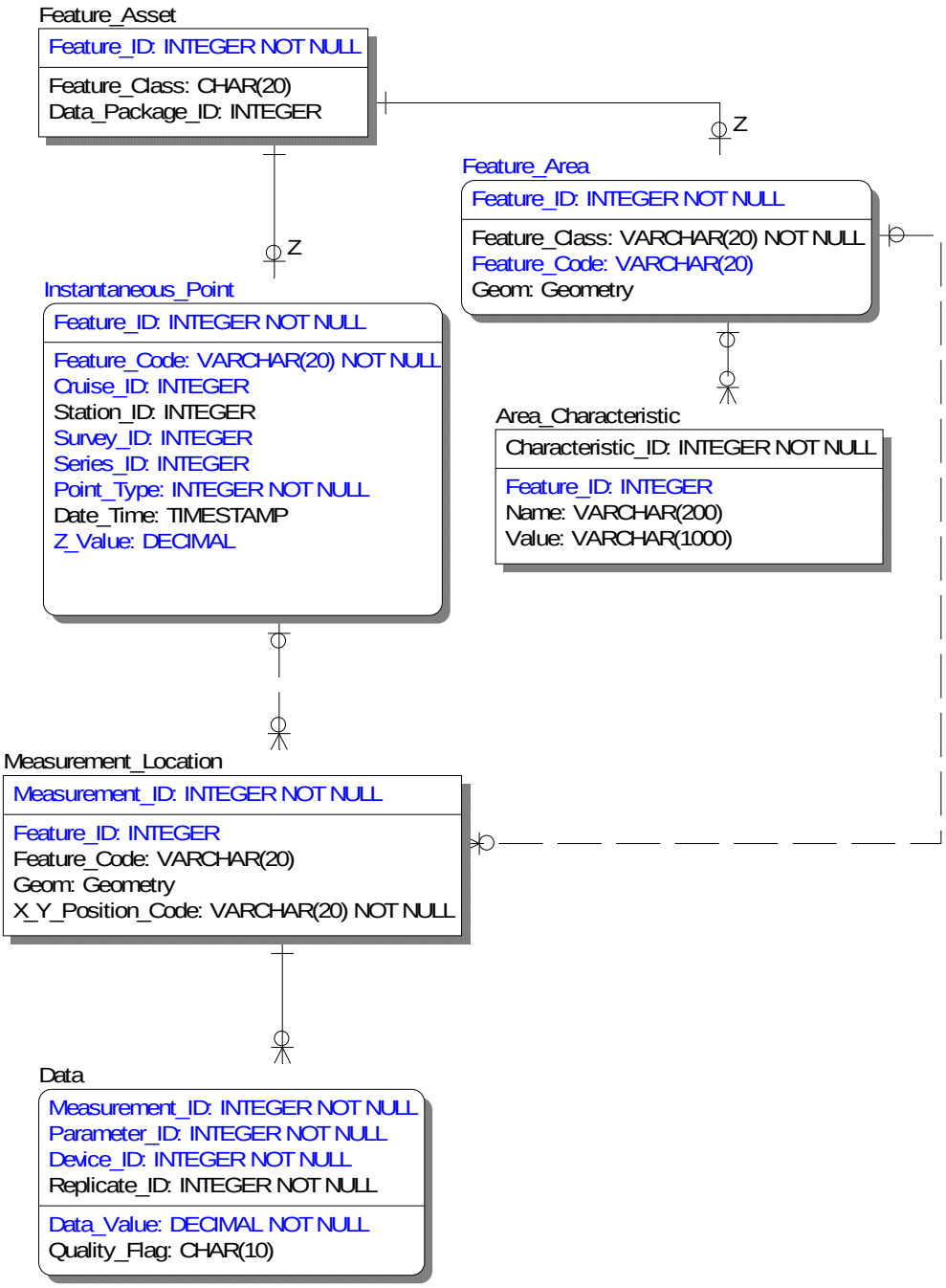

Figure 6: Feature_Area and Area_Characteristic are used to store typical vertical profile shapes. These tables are also used in defining bounding limits on profiles.

6.3.3 DND maritime operation areas

The Feature_Area and Area_Characteristic tables are also used for storing the maritime operation area names and locations (Department of National Defence (1992)). The DND operation areas are bounding polygons located throughout the Northwest Atlantic.

6.3.4 Bounding envelopes of data values

Typical shapes may be useful although shapes with assigned data values are of greater utility. Envelope profiles of environmental data can also be stored with the database. Envelope profiles are best described as bounding profiles on an environmental variable–i.e., profiles that are based on a calculation, where the measured data profiles are used to define bounds or limits on the data.

Bounding profiles may be calculated in different ways depending on user requirements (the database and data model do not perform any of these computations). As a simple example, the bounds may be defined by the minimum and maximum of the variable. In this case, the user defines an area in the ocean as a geographic area of interest. The area could be specified using a simple box, or more elaborate multi-sided polygon. Within the area, all profiles of a particular type (e.g., temperature profiles) as used to define the observed low value and observed high value for specific vertical levels in the profile. Thus, at each predetermined level (e.g., 10m, 25m, 50m, 75m, ...) the minimum and maximum observed value is recorded based on the sample of all profiles in the area. Combining all minima (and then all maxima) results in two pseudo-profiles being defined.

The REA PDB is capable of dealing with these pseudo-profiles in one of two ways.

In the first method, the pseudo-profile is defined for an area (Figure 6). In this case, Feature_Area is used to identify the *Feature_ID* of a specific area. That unique *Feature_ID* is then used in Instantaneous_Point and Measurement_Location to identify a set of *Measurement_ID* values for the feature area. These *Measurement_ID* values are then used in Data to define the values of the pseudo-profile.

In the second method, the pseudo-profile is defined for a location, in a similar way to the actual profiles. The Instantaneous_Point table identifies a unique *Feature_ID* for each minimum or maximum profile. Since the table is setup to primarily deal with cruise or asset specific data, many nonkey fields in the table will be blank (e.g., *Cruise_ID, Station_ID*) for this application of the table. This does indicate a violation of third normal form (Wikipedia (2009)). However, we consider the usefulness of having these pseudo-profiles accessible via the same table as the data to out-weight the normalization issue.

In fact, the decision to store the bounding profiles in the Instantaneous_Point table directly results in the *Cruise_ID* and *Date_Time* fields being made nullable. The nullable aspect of these fields is not a requirement in normal measured profiles. It is solely a result of the desire to place these bounding profiles within the same table. The bounding profiles are not cruise specific and thus do not have a cruise number; nor would they have a specific time in our particular example for minimum and maximum profiles as presented above.

Each pseudo-profile would have a unique *Feature_ID* in Instantaneous_Point. Each unique *Feature_ID* indicates a set of measurements in the Measurement_Location table. The actual computed values would appear in the *Data_Value* field in the Data table. The Data table would also contain numeric identifiers for the parameter in *Parameter_ID*. Using the Parameter table, this numeric would indicate the particular type of bounding profile. In a similar way, the numeric for *Device_ID* would indicate a "COMPUTED" parameter in the Measuring_Device table.

6.4 NADAS data source

The Non Acoustic Data Acquisition System (NADAS) is a serial-based communications line onboard the research vessel CFAV QUEST. The serial line is effectively the backbone through which the serial data are distributed throughout the ship. The serial line accepts all serial input from a suite of instruments. This suite of instruments can change between or even during an individual cruise.

The positional fixes from the QUEST GPS receiver are distributed using the NADAS serial line. As well, any other instruments that can report output to comply with the serial specification can also report their values over the NADAS serial line. This typically results in a multitude of data inputs all reporting values on the NADAS data stream.

It is important to keep in mind that when we use the terminology "NADAS data", we are actually referring to the data on the NADAS (i.e., on the Non Acoustic Data Acquisition System). NADAS does not produce data on its own, but rather acts as a conduit for the transport of serial-based data.

All data placed on the NADAS are date/time stamped based on the GPS signal. However, data do not necessarily have position stamps. This results in a slight mismatch between the positional information in the NADAS stream and the sensor measurements. Annex E provides more detail on the NADAS system.

The evolution of the NADAS date/time stamps is also important for this work. Although the history of the system is not well documented, there is speculation that the early date/time values originated from the PC logging the data. This is thought to explain synchronization problems between date and time values in the early 1990's. These synchronization errors would result when the PC clock was not properly set to Greenwich Mean Time (GMT), resulting in occasional offsets noted in the date/time values in the NADAS data.

Each sensor on the NADAS provides data to the system in a character based data string, in a form loosely based on the National Marine Electronics Association (2002) (NMEA) 0183 specification. Each NMEA-like string is numerically coded to indicate the source of the data. Sensor output can be added to and removed from the NADAS data stream on an adhoc basis.

6.4.1 NADAS specific tables

The tables added to accommodate storage of NADAS data are named Survey_Info and Survey_Key. These tables are shown in Figure 7 with the Track and Instantaneous_Point tables. Although added for NADAS, the Survey_Info and Survey_Key tables also provide functionality useful for other data collection activities.

Track provides the identification of lines. As noted previously, these lines can be single or multi-segment. An individual track as identified using a single *Track_ID*, can be subdivided into segments using the *Feature_ID* of the track record. Thus, an individual line can be identified using multiple unique *Feature_ID* values for a particular track.

The Survey_Info table is used to identify a particular data collection activity. This activity is assigned to a particular track via the *Track_ID* field in the Survey_Info table. The survey activity can also be assigned to specific segments of the track. This functionality uses the Survey_Key table to link a specific activity to a specific segment of the track. In this way, a particular track *Feature_ID* is then related to zero, one or many *Survey_ID* values in the Survey_Key. The *Survey_ID* in Survey_Key is then related to the detailed description of the survey, that being in Survey_Info. Finally, Survey_Info is related to the Instantaneous_Point table where a set of instantaneous points are related to a *Survey_ID*.

Within Track, *Feature_Code* is used to assign a name to a specific geospatial line. For example, a particular line that is repeated over a single cruise or over multiple cruises may be assigned a common name for identification (e.g., Med Line A). This name could be included in the *Feature_Code* field in the Track table. The Survey_Key table provides the ability to associate multiple surveys to a specific line segment (as indicated using the *Feature_ID*) that is part of a specific track.

As an example, consider a ship which transits from point A to point B, then to point C, and finally to point D. Suppose there was no data collection from A to B. Then, from B to C the NADAS system collected data types 1 and 2. Finally, from C to D the system collected data types 1, 2 and 3.

In this scenario, there are three line segments: A to B; B to C; and C to D. In terms of the Track table, these three lines could be divided into three track records. All three track records could have the same Track_ID indicating they are a single track. However, different *Feature_ID* values would indicate that the track is divided into three segments. Then the *Geom* would only contain segments of the total track.

There are also three surveys, particularly those data collection activities related to data types 1, 2 and 3. On track AB there were no surveys since no data collection activities took place. On track BC, there were two surveys corresponding to data collection for data types 1 and 2. The survey for data types 1 and 2 continues onward, to include track CD. On track CD, we also have a new survey for data type 3.

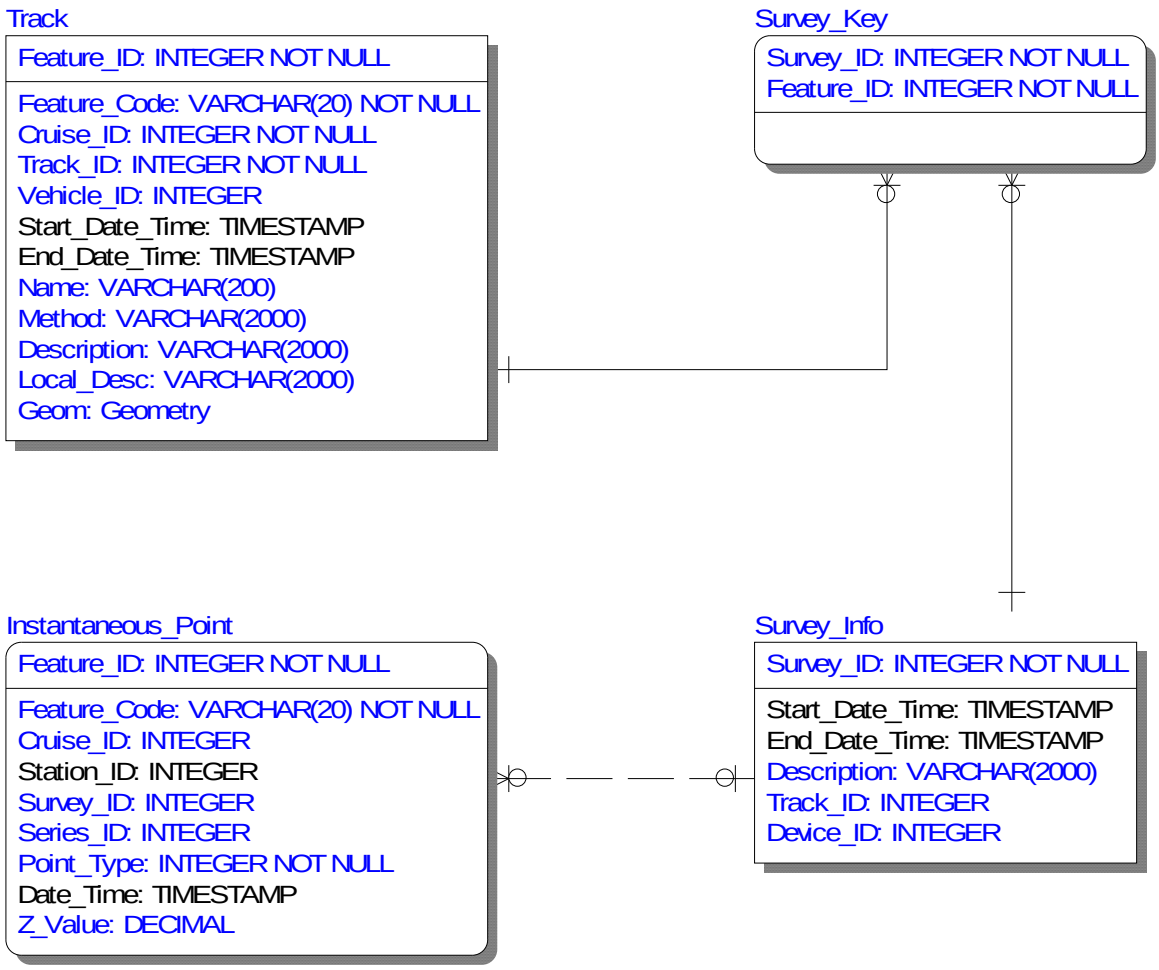

Figure 7: Survey and collection line tables used in NADAS data storage.

Each survey is described using Survey_Info. This table is used to describe more details of the specific data collection that pertains to that survey. As an example, the table structure allows a user to query the database table Survey_Info to identify a particular *Survey_ID* value. Based on the *Survey_ID*, one can then use the Survey_ID field in Instantaneous_Point to determine all *Feature_ID* values that originate from this *Survey_ID*. Then, using the *Feature_ID*, all *Measurement_ID* values from Measurement_Location can be determined. Finally, all data values corresponding to those measurements can be found in the Data table. In this way, all the data from a single survey can be identified within the database.

One might think that a single device could be defined as "NADAS" and this device included in the Device table. This would then allow surveys to be described as using this device, via the Survey_Info table, with the Survey_Key defining features of the Track table corresponding to that survey. Then, the *Survey_ID* used in Survey_Key would be used in Instantaneous_Point

with a set of point *Feature_ID* values. Although such a solution is structurally possible for the NADAS, technically this is not the proper use of the device table. This is because NADAS is not actually a measuring device but rather is a communication mechanism. Proper use of the tables would entail describing each sensor on the NADAS system in the Device table. The proper specification of NADAS data may not be possible given the lack of sensor specific metadata available for past trials.

Note that the *Feature_ID* values in Instantaneous_Point are NOT the *Feature_ID* values in Track. This is because the *Feature_ID* in Track is an identifier for the particular line segment as a track, while the *Feature_ID* values in Instantaneous_Point are identifiers for instantaneous points.

The Instantaneous_Point table then defines all the points contained within the NADAS survey. The Measurement_Location table would identify the geospatial location of the point in terms of latitude (y) and longitude (x). If depth of the measurement is known, z may also be specified. The actual data values are then stored in the Data table. The *Device_ID* in Data would be the same as used in Survey_Info.

The data values present in the original REA LDB table are actually the means (i.e., averages) of the values measured and recorded in the NADAS stream. These means are to be stored in the REA PDB. An example of this is with the relative humidity data, which is recorded from 3 sensors and reported to the NADAS stream. These three values are averaged and it is this average that is recorded in the REA PDB.

It is useful to acknowledge that no metadata exists for the NADAS data stream. In the data model, this lack of metadata manifests itself as the absence of a NADAS specific metadata table; this as compared to the XBT and CTD metadata tables.

Some NADAS data will have interpolated positions because the NADAS 013 (see Annex E) record does not exist within the record block. This results in *X_Y_Position_Code* being included in the Measurement_Location table to indicate the interpolated position.

6.5 Eastern Canada shallow water ambient noise dataset

The east ambient noise experiment reported in Hazen and Desharnais (1997) and in Franklin (1997) outlines an experiment during which ambient noise data were collected in the waters off eastern Canada. The data are actually sound pressure level measured over a particular set of frequency bands. Since there are no intentional sound sources (i.e., sources introduced as part of the experiment), the resulting sound data represents the ambient noise level of the surrounding environment.

The data were initially acquired from a spreadsheet containing data and computed values. The data were included as part of the initial load into the REA LDB. Unfortunately, there were errors in this initial load. For example, measurement fields in the original table ecs_wind_obs are not wind observations, but rather the sound pressure levels. As well, the records reporting a

frequency of 0 Hz have a data value that is a computed mean rather than a measurement. The data migration plan presented here should correct these problems.

To store these data in the PDB, we first introduce the metadata specific table named Ambient_Noise (Figure 8). This table is designed to store the particulars of each ambient noise collection site, including the site name, the general characteristics of the sediments, and a comment on shipping activity.

The ambient noise experiment consisted of 12 individual aircraft flights to acquire the data. These 12 flights are considered surveys (Figure 9), with the flight information then stored in the REA PDB table Survey_Info. The assigned *Survey_ID* is then used in Instantaneous_Point to identify each flight.

For each survey, the individual site numbers are assigned a unique *Station_ID*. There were at most four sites involved in each flight. Thus, there are four possible *Station_ID* values for each *Survey_ID*.

We also must recognize that each measurement site (i.e., *Station_ID*) has a set of sensors deployed in a triangular pattern (Figure 9). For the PDB, the site location will need to be re-specified into the three apex locations for each triangle, computed based on the site location and the triangle dimension of 10 nautical miles for each side. The site location is assumed to be at the centre of the triangle with the triangle orientation being such that an apex is due north, with the triangle base parallel to a line of latitude. This results in three locations per site, inside the table Measurement_Location. This technique of dividing the site into three points in space allows the separation of the data collected at each apex location.

The *Feature_Code* in Measurement_Location is used to indicate that the data is of type AMBIENTNOISE. A particular feature code value is added to the data model to account for this (see Annex D).

The Data table is used to store the three components of the data value. The three components are the centre of the frequency band, the mean of the sound pressure over that band, and the temporal standard deviation of the sound pressure over that band. In table Data, the *Measurement_ID* is used as a grouping value, to group all the data at a particular site. For each of these three components, a unique *Parameter_ID* is assigned with an appropriate description of the parameter in the Parameter table. For all three components, the *Device_ID* indicates a described device of SONOBUOY. The device ID could separate out types of sonobuoys if sufficient information exists to identify the type. Finally, the numeric value of the centre frequency band, mean or standard deviation is stored in *Data_Value* in the Data table.

Another table in the REA LDB, named ecs_wind_obs_vs_model, contained observed ambient noise data and modelled noise data. The content of this table is omitted from the REA PDB. This is because the observed data values are already included in the previous data loading example, and the modelled data is not included because it can be reproduced.

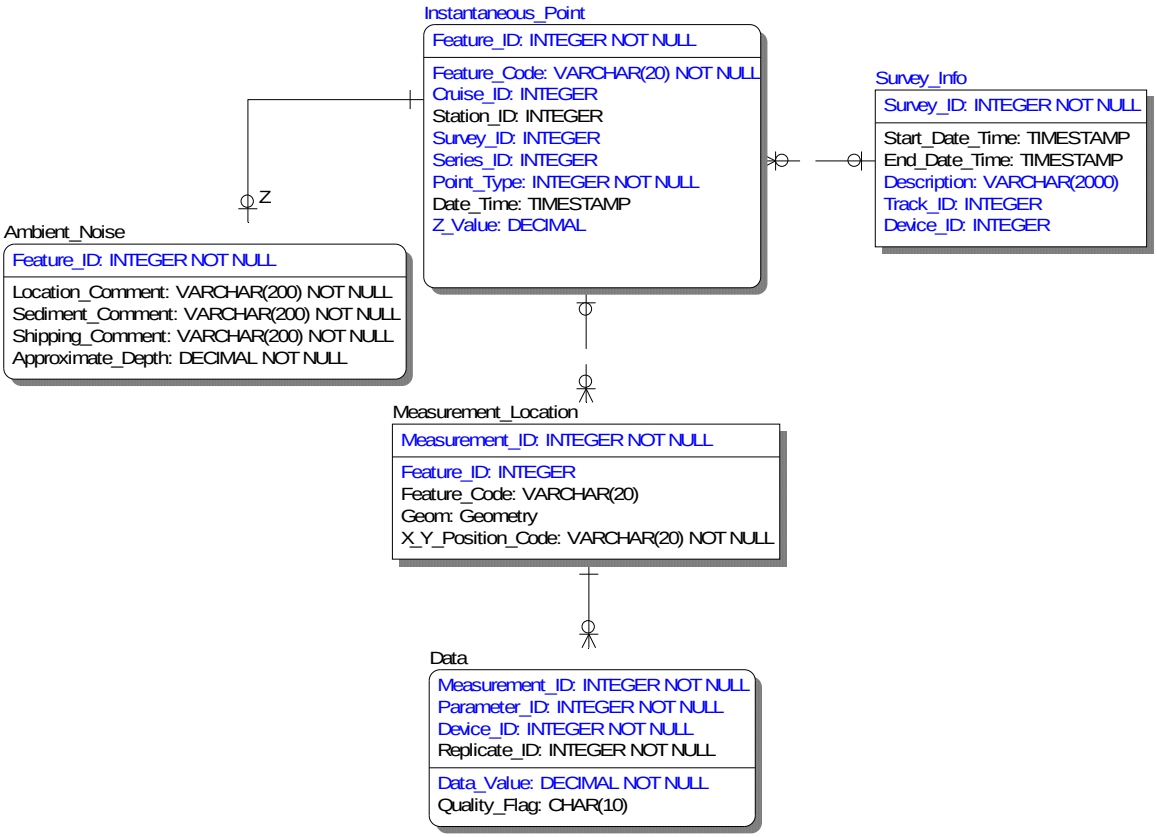

Figure 8: The storage of ambient noise data requires only the addition of one table; that being Ambient_Noise.

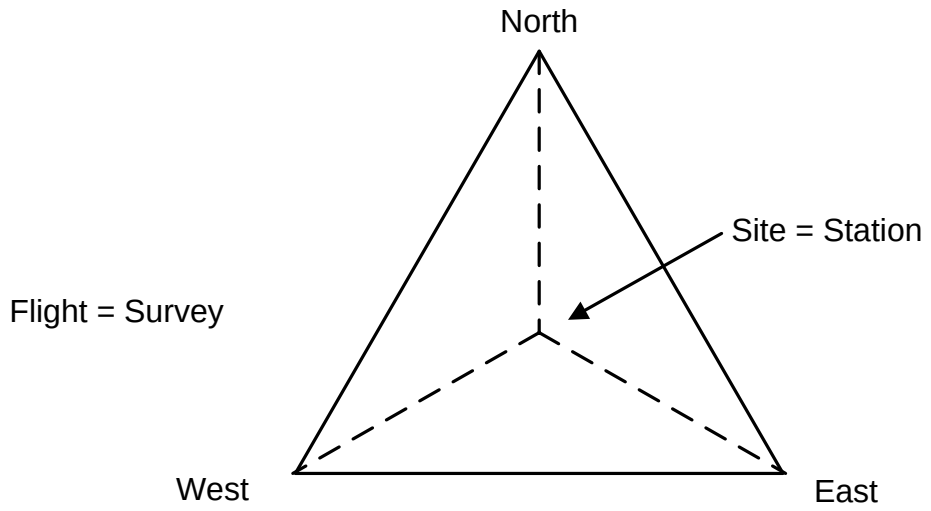

Figure 9: The terminology used in the original experiment as compared to the PDB storage. The triangle represents the pattern of instrument deployment during the experiment.

6.6 Bellhop

The Bellhop software was not functioning at the end of Deveau (2008) Phase III work and will not be considered in this work.

6.7 Transmission loss data from the shallow water database

The DRDC Atlantic transmission loss data are commonly known by the storage name of the data set: the shallow water database (SWDB). The transmission loss data are a unique data set. The data represent the loss in sound intensity as the sound propagates through the water.

In the 1980's, DRDC Atlantic (then known as Defence Research Establishment Atlantic or DREA) conducted transmission loss experiments at numerous locations in the Northwest Atlantic. A particular transmission loss experiment would begin with the deployment of a mooring line consisting of many hydrophones either in the vertical water column, in the vertical and horizontal (i.e., along the bottom) or only along the bottom (Figure 10). An aircraft or ship would then move radially away from the mooring on a particular bearing. During this steaming, the ship (or aircraft) deployed sound sources into the water column. At the time, these sources were actually explosive charges and thus became known as "shots". Each sequence of shots along a single bearing was referred to as a "run". A single transmission loss experiment could involve multiple runs (Figure 11).

The transmission loss data set has many internal relationships, and understanding these relationships is critical to understanding the placement of the data within a database structure. In an attempt to explain these relationships, we provide Figure 12. The figure shows that a single experiment (i.e., a single mooring deployment), as shown at the top level of the figure, could have multiple runs associated with it. The multiple runs are illustrated as a branching from a single experiment, with the runs indicated r_1, r_2, etc. Each specific run could have multiple shots. Note that a single shot (denoted "s") occurs at a single range, as denoted by "R" in Figure 12. That particular range occurs at a specific geographic location denoted [x,y]. For each shot, all hydrophones deployed on the mooring receive the acoustic signal. A specific hydrophone, denoted by "h", is physically located at a unique location as illustrated by the $[x,y]_m,z_n$ with m indicating the mooring location, n indicating an integer between 1 and H (the total number of hydrophones) and z indicating the hydrophone depth. For each hydrophone, a set of sound frequency bins (denoted "f") are used to measure transmission loss (denoted "T").

The data collected at the hydrophones represent the drop in received sound level as compared to the intensity of the initial sound source (Chapman and Ellis (1998)). The data are range and frequency dependent. Each shot introduced at a particular range, is used to determine the sound transmission loss as a function of frequency and depth. The data were then represented in a matrix of transmission loss data values (in dB) with the matrix columns indicating the centre frequency of the measurement band, and the rows representing the range of the shot from the moored hydrophones. A portion of a matrix is shown in Figure 13. The original system which used this matrix structure was known as the shallow water database, and is documented by Wycove Systems Limited (1985) and Wycove Systems Limited (1986).

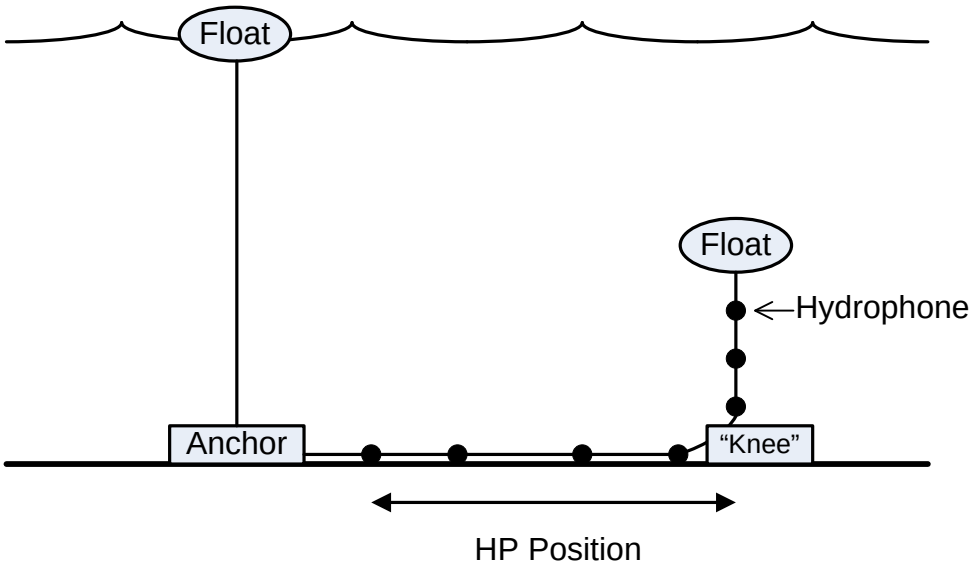

Figure 10: An illustration of a mooring deployment used for a transmission loss experiment. HP position indicates the horizontal distance from the "knee" to the hydrophone position on the array. The hydrophones in the vertical would have zero HP position values.

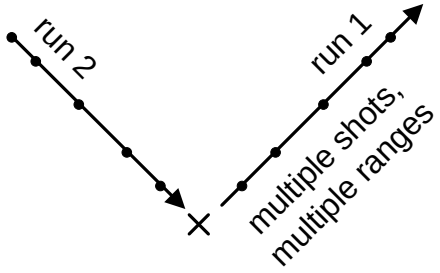

Figure 11: The mooring location is shown with an "X". The run is a line at a particular bearing. The shot locations are illustrated with black dots, and represent the locations where a sound source was introduced into the water column. Note that run 1 deployments occurred while moving away from the mooring location (i.e., referred to as OPENING) while run 2 deployments occurred while moving toward the mooring location (i.e., referred to as CLOSING).

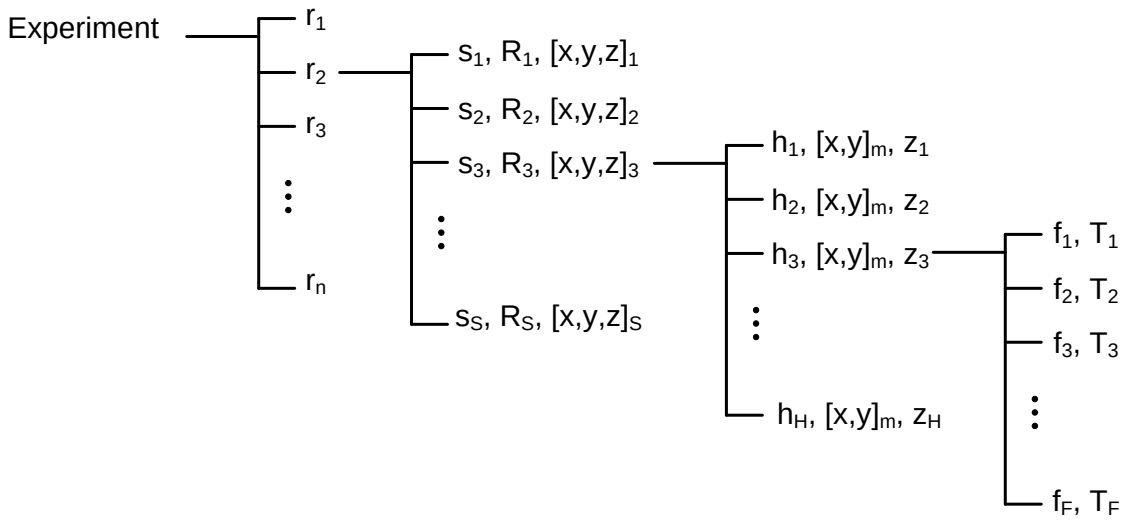

Figure 12: The relationships that exist for the transmission loss data. The figure is described fully in the text. r:run; S:shot; R:Range; $[x,y,z]_s$:the position of the shot; h:hydrophone; $[x,y]_m$: the position of the mooring; z_H: vertical position of hydrophone m; f:frequency; T:transmission loss; S:maximum number of shots on the run; H:total number of hydrophones on the mooring; F:number of frequency bins.

11/3 OCTAVE CENTRE FREQUENCIES CHANNEL 17 S/N = 3 DB,Q144,RUN 2,200 FT,CIRCULAR

		3.2	5.0	6.3	8.0	10.1	12.7	16.0	20.2	25.4	32.0
SHOT#	RANGE(KM)	PROPAGATION LOSS									
S02019	5.50	321.3	84.2	84.2	87.0	93.4	104.2	106.4	106.9	106.4	110.4
S02021	25.60	80.7	86.5	87.1	89.8	93.8	101.9	106.6	108.8	108.8	109.4
S02022	35.30	81.0	86.0	85.8	89.1	97.3	105.2	104.8	108.6	109.1	114.9
S02023	44.70	80.9	84.9	85.0	87.7	93.3	97.3	101.8	104.8	106.1	105.9

Figure 13: A portion of the original matrix input file that contains the transmission loss data. As illustration, shot S02022 occurs at a range of 35.3 km and has a transmission loss of 104.8 dB in the frequency band of 16.0 Hz. Note that run and cruise number are provided in the top line. Also note channel number. In the header information from the original input file, this is referred to as HP NUMBERS or WIRED POSITION. The channel number uniquely identifies the hydrophone for this particular mooring arrangement.

There is also a considerable amount of metadata associated with each run. Transmission loss is influenced by water depth, the composition of the ocean bottom, the speed of sound in the water column between shot and mooring location, sea surface roughness (as influenced by sea state[1] and wind speed), and sound source frequency (Wycove Systems Limited (1986)). These dependencies mean there is considerable environmental metadata associated with each run.

There are numerous complications associated with these data and the storage of these data within the existing REA LDB (version 3b). First and foremost, the actual transmission loss data do not exist in the REA LDB version 3b in parsed form. The initial matrix load into the LDB only exists in a line-by-line representation of the input file. In this form, the data cannot be easily associated with the metadata of the experiment. This is because the metadata exists on different line numbers in the table. To solve this, the import of the transmission loss data should be redone with the data loaded directly from the initial matrix tables to the REA PDB.

Although a reload is recommended, we nevertheless complete the mapping between existing REA LDB table content to the PDB table content. This provides completeness and assists understanding of how these data are treated within the LDB.

In the PDB, these data will be treated as a set of associated point measurements. For the case of the mooring, the metadata are to be stored in the Instantaneous_Point table. The *Point_Type* field identifies the type of point as being a LocationSeries (this is a value for a field; see Instantaneous_Point field descriptions in Annex C). This indicates that a series of points are associated in some way. The *Feature_Code* should indicate TRANLOSS for transmission loss. The actual location of the mooring will be stored in Measurement_Location. There will be no data in the Data table associated with this point. This is because the hydrophone mooring location has no data without a shot. The *X_Y_Positon_Code* should indicate that this position is a measured position. The x,y position is the actual mooring location, while the z value should be set to the bottom (i.e., station) depth at that position. Since a single z value is associated with the mooring, the station depth is considered more representational as compared to a zero value.

A single run is considered a single survey. The Survey_Info table is used to store the start and end date of the survey line (i.e., of the run). Note that the start date/time should be associated with the deployment of the first shot; while the end date/time is from the deployment of the last shot. The *Description* field in the Survey_Info table is used to store the area name for the experiment. As well, *Description* is used to describe if the experiment is based on ship or airborne assets and the run number associated with the experiment. Knowing the deployment platform type is critical when attempting a consistency check of the range data to the start and end times of each run (i.e., each survey). The survey is linked to a track via the *Feature_ID* field in Survey_Key. The Track table can then be used to identify all line segments associated with this set of *Feature_ID* values. Obviously, the segments in Track can be plotted to show the lines which form the runs. Multiple records in Instantaneous_Point table will also contain the same value in the *Survey_ID* field value thereby indicating the same run.

[1] The SWDB sea state code has a set lower/upper limit of 0/6. The origin of this code scheme is unclear. Documentation refers to the coding as “international”. The only international sea state coding scheme is from the World Meteorological Organization., Table 3700, which varies from 0-9. WMO wave height table 1555 also varies from 0-9. It is possible WMO table 3700 was being used, but that conditions in the 7-9 range where not conducive to experiments. The code range of 7-9 indicates wave conditions exceeding 6 m.

The *Station_ID* field in Instantaneous_Point will be used to represent the actual shot number. The shot drop locations will be indicated using the Measurement_Location table. Note that the positions will need to be computed using the mooring location, the shot range and bearing. The z position should be the depth at which the shot detonated, commonly called source depth. The *Feature_Code* for these data will indicate TRANLOSS. The *X_Y_Position_Code* should indicate that the position data is computed based on range and bearing from a known location.

The data stored for each shot drop will consist of numerous values all associated via a common *Measurment_ID* value in the Data table. The store data should consist of:

- range (in km),
- bearing (in degrees true),
- frequency (in Hz), and
- transmission loss values (in dB).

Note that the range and bearing values are in one sense, duplicated data. This is because the range and bearing are implicitly available through the series of location positions in the Measurement_Location table. As well, bearing is being stored in the Transmission_Loss table as metadata associated with the experimental run. However, we maintain these duplicate data as a means of providing direct access to the range and bearing data, rather then via the point geometry type. For all data records, *Replicate_ID* will be "1" and *Quality_Flag* will be "unknown".

Finally, we consider the hydrophone specific metadata: the hydrophone depth (i.e., in the input header record known as HP DEPTH), the hydrophone numbers (i.e., in the input header record known as WIRED NUMBERS or HP NUMBERS or CHANNELS) and the hydrophone positions (i.e., in the input header record known as HP POSITION). All of these metadata provide information on the placement of the hydrophones on the array mooring (Figure 10). The hydrophone depth indicates the depth from the water surface to the hydrophone. The hydrophone position indicates the horizontal displacement of the hydrophone from the knee of the mooring. Finally, the wired numbers (as known as HP numbers or channels) indicates the physical wiring arrangement made on the hydrophone array. Each hydrophone had an associated jumper wire as connecting the hydrophone to the larger array. This connection is numbered and accounted for via the wire number. All of this metadata are specific to the experimental layout of the mooring.

These metadata will be stored in the Device_Class_Detail table. Each metadata class will be given a *Descriptor* of hp_depth, hp_number, or hp_position. The *Value* field will be filled with the numeric value corresponding to that particular hydrophone on that particular mooring. The *Device_ID* number will be linked back to the Measuring_Device table. In Measuring_Device, the *Name* will be "hydrophone". A unique *Device_ID* must be assigned for every hydrophone, on a per mooring basis. This is because each hydrophone on each mooring has a unique depth, number and position.

An additional table is added to store the associated transmission loss metadata. The table, Transmission_Loss, contains metadata specific to an individual run. Much of these metadata are actually stored in the existing REA LDB. However, it is unclear if the process to load the

PDB would benefit from direct access to these data in the LDB. Again, it may be more straightforward to load the data directly from source files.

6.8 Gridded data

6.8.1 Dalhousie temperature-salinity climatology

Gridded data refers to any data set that occurs at a regular or near-regular spatial interval. This interval can be in the horizontal, in the vertical, or in both horizontal and vertical dimensions. In a gridded data set, the data represents measured or computed values over a spatial area or volume.

Gridded data is common in oceanography for either measured data that have been processed to a specific grid or data produced by a numerical model. In particular, modelled data often occur on a regular grid because of the discrete model mesh from which it originates.

There are several gridded data sets that exist within the REA LDB. Bathymetry tables from numerous sources are one such example (more on bathymetry in Section 6.10.2). However, in this section we are concerned more with gridded environmental variables and possibly model output.

An example from the existing REA LDB would be the temperature-salinity Dalhousie (TSD) gridded data set. These data are gridded to 15 depth levels once per month, covering an area approximated by the Scotian Shelf. The TSD data set as it exists in the REA LDB is structured in tables that take advantage of the object nature of Postgre. The table tsd_geopoints contains the depth information, in double precision array structure (i.e., an array object). This means that the depth data are not stored in real or float numeric. Rather, the data are stored in an array object that consists of a set of real numbers. In other words, a single field value is actually an array of values. The Postgre environment is capable of understanding and manipulating this array object. However, if the table is linked as an external ODBC data source (e.g., via another DBMS like Microsoft Access), the object is not understood. In the Access case, the object is converted to a simple text string. The object then loses the additional functionality that was present in the Postgre environment.

This is an important point if access to the PDB is likely to occur via other software such as Microsoft Access or the ESRI Arc products. These products will not be able to properly interpret the objects used within the Postgre environment. If the data were stored using simple structures (e.g., reals, floats, characters) the external software could access and manipulate the data. For example, ESRI provides the capability via ODBC connections to access a database on a remote computer. Once access is established, individual tables from the database can be used as layers in the ESRI environment. A query definition can then be created for the specific table which then restricts the query return set. This process is described by Isenor (2008).

The redesigned tables in the PDB store the grid data in decimal form, not utilizing the Postgre object type. The portion of the production data model dealing with gridded data first recognizes that two types of data may exist: scalar or vector quantities. The table Scalar_Quantity stores a

decimal scalar with a date/time stamp. Each scalar quantity is assigned a *Feature_ID*. Similarly, Vector_Quantity is used to store the vector components of the grid value. For both tables, the *Feature_ID* is used to link to a unique position on a grid, here referred to as a mesh. The table Mesh_Point defines a particular mesh point on a particular mesh. Multiple meshes can be defined using the Mesh table.

As an example, consider a multi-meshed numerical model producing output temperature over 10 depth levels. In this case we use Mesh to define each of the grids, noting the number of grid cells in the x,y,z dimensions. For each mesh, we also note the number of active points in the grid using the *Total_Points* field. Next, for each cell in each grid we define a *Feature_ID* in Mesh_Point. We can locate this cell either in i,j,k space, or in terms of an x,y,z point on the physical Earth using the *Geom* field. Each scalar or vector quantity is defined in Scalar_Quantity or Vector_Quantity, with the *Feature_ID* in those tables linking back to the *Feature_ID* in the Mesh_Point table.

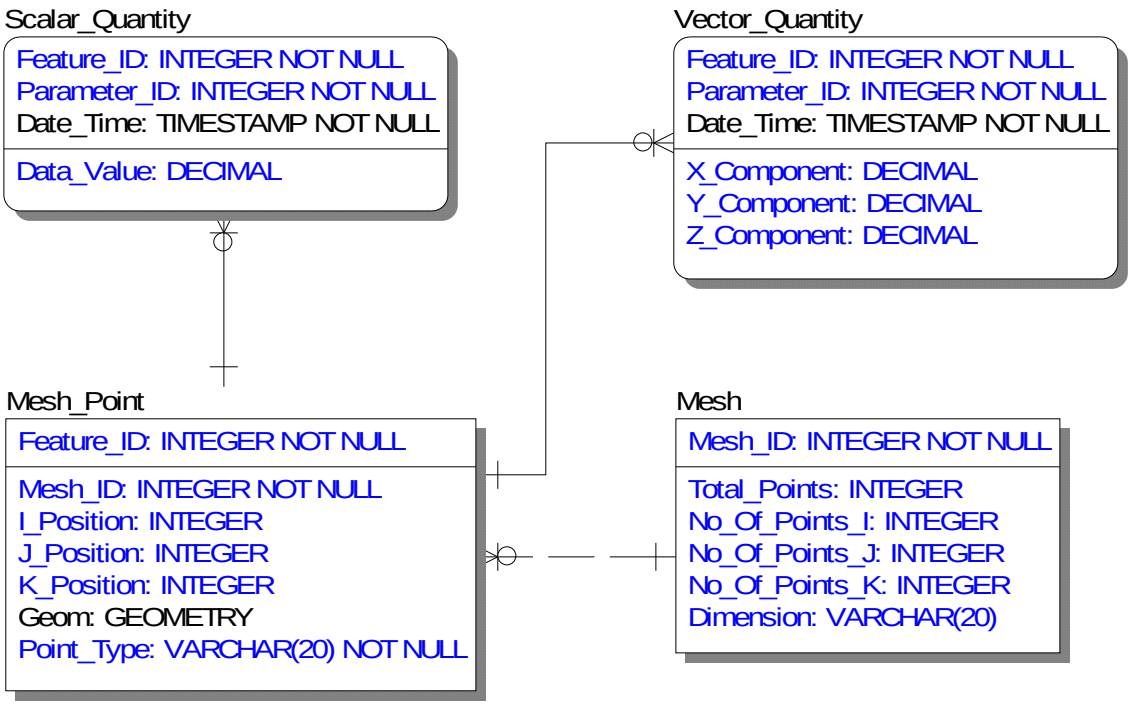

Figure 14: Gridded data are stored using the mesh tables.

6.8.2 Sediment Thickness

The sediment thickness in the REA LDB was obtained from Divins (2006) and is based on data from the National Geophysical Data Centre (NGDC). These data will also be stored in the mesh tables introduced in section 6.8.1. These data are on a 5-minute grid and originate from a suite of sources including the Ocean Drilling Program (ODP) and the Deep Sea Drilling Project (DSDP). More details on these data can be found in Annex F.

In the LDB, these data are stored in table sedthick. The latitude and longitude position of the grid is presently in *the_geom* field, and these data should be mapped and stored in the Mesh_Point table. The *Geom* field in Mesh_Point will be used to store the geometry for the x,y positions. The actual data will be stored in Scalar_Quantity table. The *Feature_ID* field is used to link the data to a particular mesh point. The *Parameter_ID* is linked back to the Parameter table. The *Date_Time* field in Scalar_Quantity should contain a single date/time value appropriate for the entire data set. The suggested date/time value is taken from the download time of the input data file: July 24, 2006 at 15:59:00.

6.9 Scotian Shelf Sediment Data

These sediment data were reported by Mackay, *et al.* (1986) and include a total of 56 measurement locations on the Scotian Shelf, in the area south west of Sable Island. The data include the number of measurements at a specific location, the mean velocity (kilometres per second) of the acoustic signal in the bottom, the error estimate associated with this velocity (kilometres per second) and the average clearance of the equipment above the seabed during the measurements. There are also high cluster and low cluster values which indicate the number of measurements clustered in a group and existing beyond one standard deviation from the reported mean velocity value. These data were originally stored in the LDB table named sediment_ss_position.

Once again, these data are stored using the tables Instantaneous_Point, Measurement_Location and Data. There are 56 records of data in the original source data set. These 56 point measurements are described with 56 records in Instantaneous_Point using 56 different *Station_ID* numbers. A single *Survey_ID* can be used to group the data into a single survey. As well, a single *Feature_Code* can be used to indicate the type of data, that being SEDIMENT.

The 56 individual measurement locations are described in Measurement_Location. Again, 56 records will exist in Measurment_Location. A unique *Measurement_ID* will be specified for each record. This *Measurement_ID* will be used in Data to group all data values. Thus, in Data a set of up to seven parameters (group, number of measurements, mean velocity, error, high cluster, low cluster, and average clearance) will contain a single *Measurement_ID* value. When no high or low cluster values exist, the number of records in the *Measurement_ID* set will be reduced.

6.10 External data sets – the implementation of user exits

External data sets represent the last category of data currently held within the existing REA LDB. These data sets are represented by the bathymetry data, and the numerous tables originating from the Geosciences Canada Atlantic CDROM for the Scotian Shelf Geoclutter Project. It should be noted that the Geoclutter tables presently in the LDB are representations of the original shapefiles from the Geoclutter CDROM.

Many other data sets included in the PDB could also be dealt with as an external data set. The Scotain Shelf sediment data, the sediment thickness data, and the Dalhousie temperature-salinity climatology are all external data sets. For these cases, a conscious decision was made to include the data sets directly into the PDB. This is because of the smaller sizes of the data sets and the expected frequent usage of these data sets.

User exists are introduced as a means of managing external data assets. User exits extend the functionality of the REA database, allowing external data sources to be incorporated within the REAS without incorporating the data asset within the REA PDB. User exits may be developed within or outside the REA PDB and provide support facilities such as data extraction, product generation, access to external data sources from within the REA PDB, and so on. Where a user exit is implemented it can have a significant effect on the complexity of the task and the overall performance of the end product.

A user exit is the process of creating a link between an external data asset and the REAS. This link allows access to the external data asset via the PostgreSQL environment. The management of this link is controlled by the REA PDB. Effectively, the PDB is used to direct the processing software to the external data asset, and to control which processing software is used to acquire the data asset. The linking negates the need to store the data directly in the PDB.

The process of developing a user exit is described in Figure 15. Each step is numbered and is described below.

1.0 Start

2.0 Identify user exit requirements: Before implementing a user exit, it is important to understand the requirements it is meant to address. Attempts should be made to reduce the number of disparate technologies used in implementing user exits (i.e., select a technology that has robust tools).

3.0 What type of user exit is required?: The first step to implementing a user exit is to determine the specific type of user exit that is required. Factors which will affect the type of user exit include: 1) where the user exit will be accessed from, 2) the underlying technologies used to enable the user to develop the required functionality, and 3) interoperability required with other systems and user exits.

4.0 User exit developed within the REA PDB: PostgreSQL includes robust geospatial query capabilities, in addition to standard SQL extensions. User exits that use data within the REA PDB can be significantly more efficient when implemented within the REA PDB as well. This

approach reduces unnecessary overhead associated with processing requests over a network or through a data access technology such as Java database connectivity (JDBC).

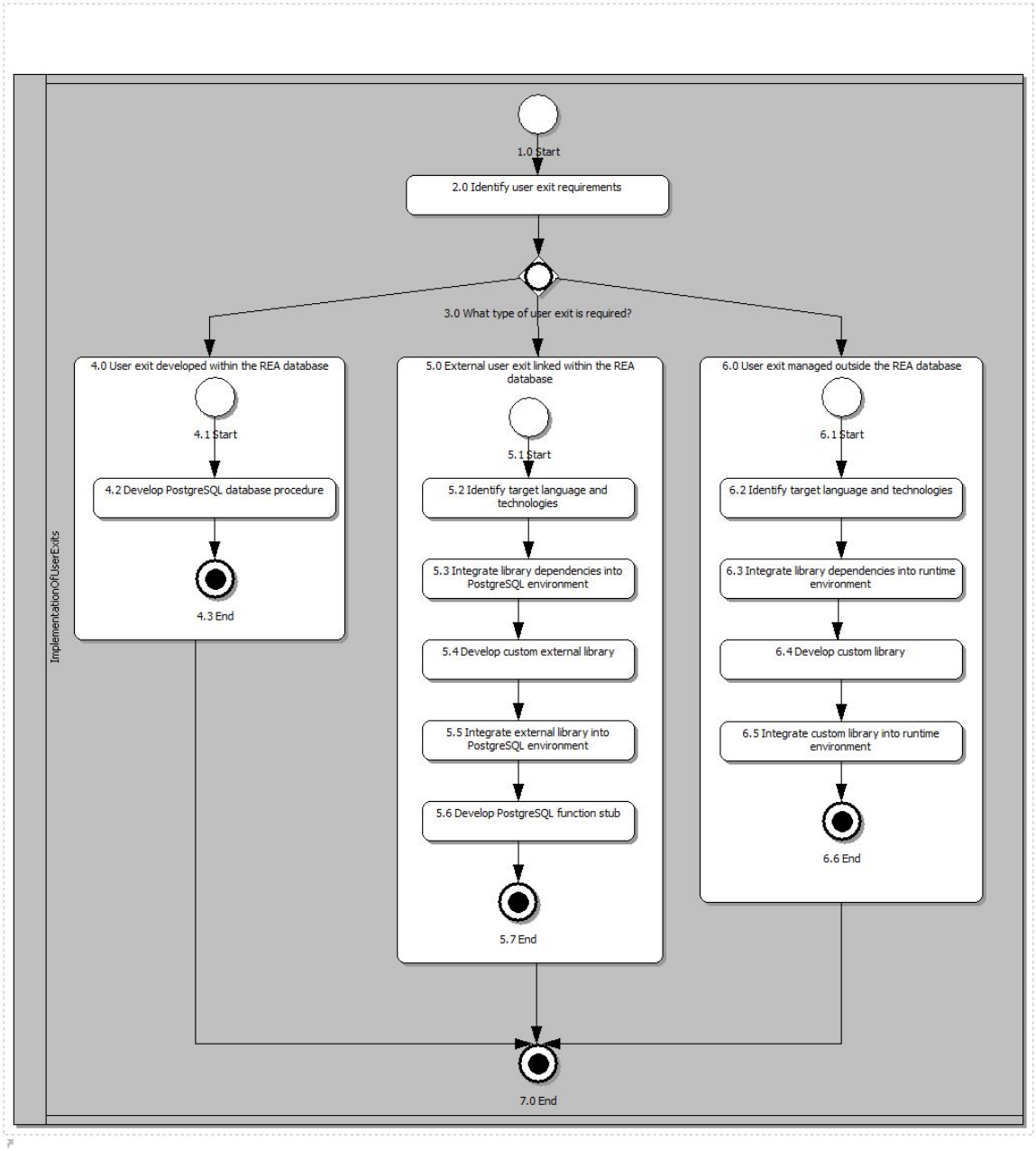

Figure 15: A graphical flow chart of how to create a user exit. The process is divided into three streams, depending on where the user exit is managed.

Some disadvantages of this approach include the proprietary nature of vendor specific procedural languages and the limited interoperability of these user exits with external data sources.

4.1 Start

4.2 Develop PostgreSQL database procedure: In the context of the REA PDB, development of the user exit follows standard PostgreSQL stored procedure development process. In this case, the procedure to access the external data is stored directly in PostgreSQL procedures.

4.3 End

5.0 External user exit linked within the REA PDB: External user exits linked within the REA PDB are generally implemented as libraries that are called from function definitions (or stubs) implemented within the REA PDB. This approach provides robust functionality by allowing the user to take advantage of application programmer interfaces (APIs) which provide functionality not inherent to the REA PDB.

Advantages of this approach includes a reduction in the amount of physical programming required by allowing the developer to use pre-built functionality in existing APIs and the functionality can be reused when implemented in a library.

Disadvantages of this approach include the additional effort required in order to make external functions available from within the REA PDB (i.e., development of function stubs) and the affects on performance when making calls to external libraries.

5.1 Start

5.2 Identify target language and technologies: The choice of target languages and technologies will depend upon the functionality required and the support from within the REA PDB. Java and C are two key languages supported by PostgreSQL, the choice of which will depend upon the APIs required in order to provide the desired functionality.

5.3 Integrate library dependencies into PostgreSQL environment: When using external libraries, they must be integrated into the PostgreSQL runtime environment in order for user defined functionality to be available. Integration typically involves either adding it to the server path setting, or placing the libraries into a specific location within the PDB server directory.

5.4 Develop custom external library: User exits not created directly within the PDB are usually implemented as libraries which are subsequently integrated into the PDB environment. When developing custom libraries, care should be taken to ensure that they contain only the capabilities needed from within the PDB environment.

When external libraries are integrated into the PDB environment, all public functions in the library are typically available and can be exposed as functions within the PDB. For

example, if one develops a library to convert between units of measure meant for use from within the PDB, but also chooses to include a function which executes user defined operating system commands, one could expose this command line execution function and allow potentially damaging and unintended side effects to occur from within the PDB.

5.5 Integrate external library into PostgreSQL environment: When dealing with libraries used to extend PDB capabilities, their integration into the PDB environment is typically more complex than simply installing a library on a server. The additional complexity is due to security and integrity of the PDB runtime environment. For example, a library which provides command line execution facilities would not be integrated within the PDB environment. It is the responsibility of the server administrator to understand the functionality being provided by the external library. Furthermore, a library which provides conflicts with the underlying PDB environment should not be integrated.

Developing libraries with very focused capabilities and with the target environment in mind will typically not encounter the issues described above.

An external library accessed from within the PDB environment is more tightly integrated into the PDB installation whereas its dependencies are usually handled as any normal library installed into the server operating system. A typical approach to dealing with user defined libraries is to force the contributor to place them in a specific location (which would normally only be writable by the server administrator).

5.6 Develop PostgreSQL function stub: A function stub is simply a user defined PDB function or procedure that calls a function in the external library. There can be limitations on how the functions are defined (depending upon the database vendor) and the developer should be aware of these before developing the external library. Once the function stub is implemented within the PDB, the function in the external library will appear as a SQL function.

5.7 End

6.0 User exit managed outside the REA PDB: User exits managed outside the REA PDB provide maximum flexibility with local and remote databases and can use custom APIs to take advantage of pre-built functionality.

6.1 Start

6.2 Identify target language and technologies: The choice of target languages and technologies will depend upon the functionality required and the support, but without requiring access from within the REA PDB, a greater variety of languages and technologies are at the disposal of the developer. The choice of languages will depend upon the APIs required in order to provide the desired functionality.

6.3 Integrate library dependencies into runtime environment: In this scenario, integrating library dependencies is the same as installing dependencies for any application. One

must take care that the dependencies do not conflict with the PostgreSQL environment (if installed on the same server).

6.4 Develop custom library: Development of the custom library follows the process for developing libraries targeted for the operating environment of choice. The approaches used and functionality provided are less constrained than the development of libraries which must be accessible from within the PostgreSQL environment.

One must still take care that the library has no adverse effects on the PostgreSQL environment if installed on the same platform.

6.5 Integrate custom library into runtime environment: Integrating the custom library into the runtime environment follows the standard practice of deploying libraries to the operating environment of choice.

6.6 End

7.0 End

6.10.1 User exits compared to uDig

It is useful to understand the relationship between uDig (described in section 5.2) and the user exits described in this section. This will also be helpful for understanding the purpose behind user exits and when those exits may not be required.

As stated previously, a user exit is a procedure for allowing access to external data sets via the PostgreSQL environment. The fact that access is via the PostgreSQL environment is a critical aspect of the user exit. This type of access means the data are obtained and manipulated from within PostgreSQL.

Other types of access that do not utilize PostgreSQL database are also possible. For example, uDig is capable of using data stored in shapefiles. As well, the software is capable of accessing the data stored in a PostgreSQL database. uDig could then be used to query or join the two data assets and provide a graphical output based on the query.

To extend the example, we consider the use of shapefiles from the Geoclutter project. In the simplest case, the shapefile may be accessed via uDig. In fact, Figure 2 shows uDig accessing two Geoclutter shapefiles for bedrock outcrops and surficial geology. For this type of access, the shapefiles do not need to be assembled within a user exit. Rather, the shapefiles need only be placed on a computer disk that is accessible from uDig. In a sense, uDig takes the form of a pre-built user exit, as implemented in the right-hand stream of Figure 15 .

The usefulness of the user exit as described in Figure 15 are for those users requiring access to the shapefile data within a DBMS system. Note that for Figure 2, the user does not have access to the shapefile data within the DBMS; rather they have access within the uDig software. The requirement to have access from within the DBMS could be present for tactical decision aids or

other models that require direct access to the data stored in the shapefiles. The stored functions that make up the user exit can then be used by multiple applications, thereby extending the functionality of the database.

6.10.2 User exits for bathymetry data

The user exit that could be implemented for bathymetry data is somewhat simpler in concept as compared to the shapefile example. For bathymetry, we can consider a 1-dimension array of depth values with a known grid size in the x and y directions; or we may consider a 2-dimensional array of depth values with known grid spacing. Again, the user exit would be initiated from within the PostgreSQL environment, accessing one or more points from the data file. Effectively, this allows the querying of the data file for a point depth or an array of depth values covering an area. This user exit could be implemented using either the central or right hand panel in Figure 15.

As with the example in section 6.10.1, the critical aspect of this user exit is understanding the requirement. Do we need the data accessible from within the PostgreSQL DBMS? Do we only require it be accessible from an application such as uDig? Questions like these provide guidance to implementing the user exits.

7 Processing Lineage

Lineage is the term that describes the provenance or processing history associated with a data set. Lineage information is actually a form of metadata which describes the source and business processes used to produce or modify the data set. Lineage metadata keeps an account of the processing history from data collection to final product. The REA PDB is capable of storing lineage metadata within the actual database structure.

This historical processing information is an important component of the quality of the data set. The history provides data set users with the ability to assess the completeness of the processing, and identify reputational issues with those involved in the processing. Reputation is noted by Adams, *et al.* (2003) to be a factor in the development of trust in automated systems.

The lineage model presented here represents a subset of the ISO 19115 model for lineage metadata and is shown in Figure 16. The attribute comments provided in Annex C indicate the relationship between attributes and the ISO 19115 standard.

The lineage begins with the *Lineage_ID* for unique identification, within the table LI_Lineage. The actual process steps that have been applied to the data are described in table LI_Process_Steps. The processing steps are numbered using the unique identifier of *ProcStep_ID*. Each process step has a *Description*, a *Rationale*, and a *Date_Time* of application.

The person responsible for the individual processing steps can be identified in table CI_Responsible_Party. The table JO_Process_Step_Processors is used to join or associate the person performing the processing with the processing step. Another join table, JO_Process_Steps_Lineage provides a link between a described processing step and the lineage of a data set. The join tables plus the LI_Process_Steps table provide all the content of the ISO 19115 element LI_ProcessStep (element #86).

LI_Lineage_Sources provides information regarding individual data sets in the data package. This table is the equivalent to ISO 19115 element #92. The unique identifier *Lineage_Source_ID* represents a single source of data. The sequence of *Lineage_ID* identifiers in this table (LI_Lineage_Sources) then represents a series of alterations to the data. Each source, as represented by a single *Lineage_Source_ID*, has a *Description*, *Scale_Denominator* and identifiers for the reference system, citations and geospatial and temporal extent.

The CI_Citation table in the structure allows the storage of the metadata associated with a report, article or paper. Including the citation to papers within the database allows the tracking of both data sources and data products. In some cases, data sets in the database originate from a research paper. In other cases, data from the database are used to generate a research paper. The knowledge of such publications can be incorporated into these tables. The paper metadata includes the *Title*, *Alternate_Title*, the *Reference_Date*, edition information, the form of the presentation material, citation details, and international standard numbers for books or serials. This set of metadata collectively mimics ISO 19115 element #359. The join table JO_Cited_Responsible_Party is used to link authorship to the referenced publication.

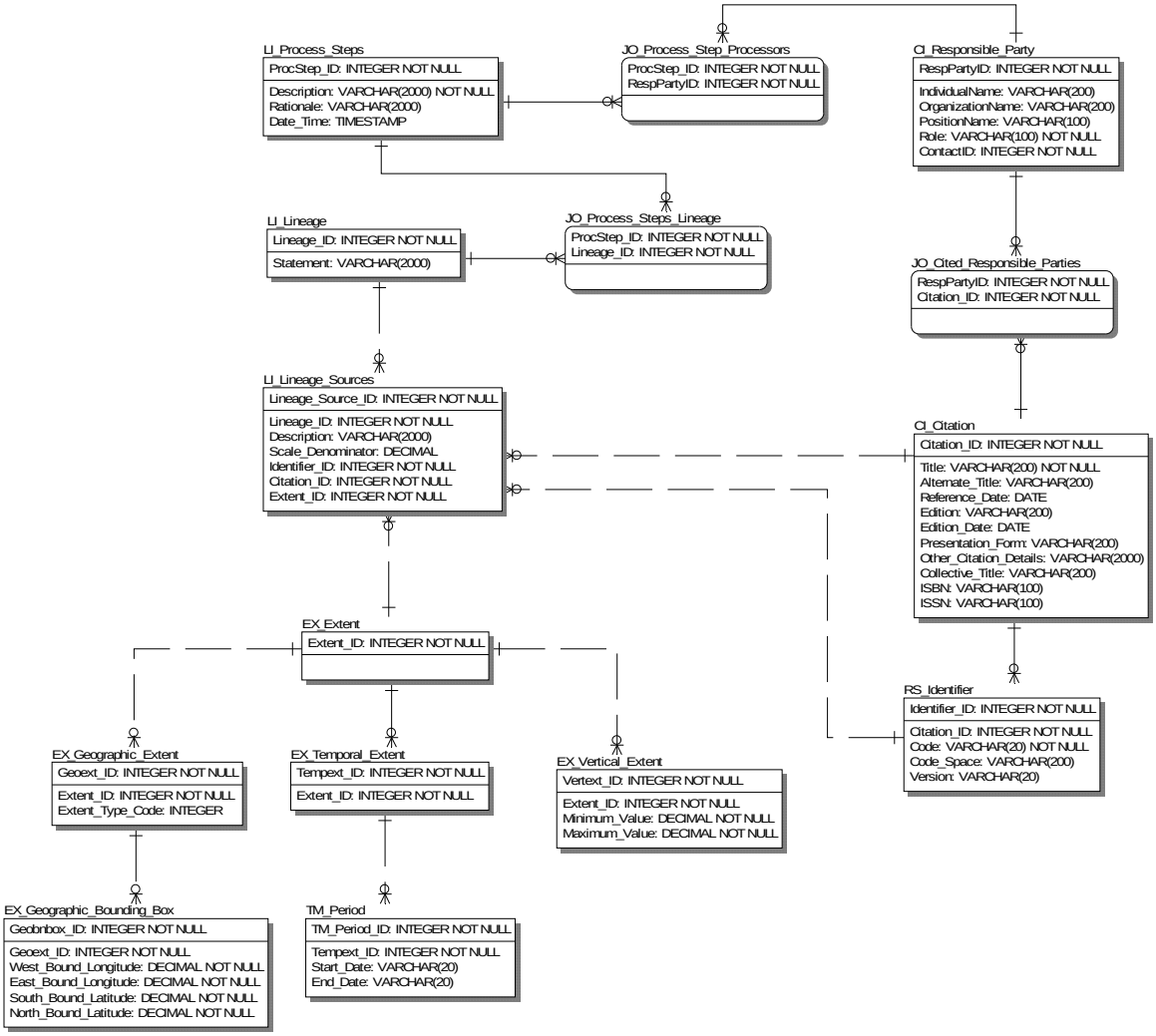

Figure 16: The processing lineage can also be tracked within the data model, with the addition of several lineage tables. These tables are based on the ISO 19115 metadata standard.

The horizontal, vertical and temporal extent of the data set can also be described using the EX_Extent tables and sub-tables. This portion of the model follows ISO 19115 element #334. The spatial extent of the data set is contained in a latitude-longitude bounding box in EX_Geographic_Bounding_Box. The temporal extent as contained in TM_Period represents a slight deviation from the ISO 19115 standard. This is because the ISO 19115 specification for time follows the ISO 8601 standard. The flexibility of date formats in ISO 8601 makes it difficult to follow in a database structure. Thus, we recommend compliance with ISO 8601 but do not enforce compliance via DBMS rules. Finally, the EX_Vertical_Extent table identifies the minimum and maximum vertical values for the data set. It should be noted that specifying the vertical coordinate reference system is not included in the current design.

8 Business processes of the REA PDB

The business processes of a database are those functions which would be conducted on the database during normal operation and maintenance. For the REA PDB, there are five primary business processes. The following describes four of these processes. The fifth process was user exits, and was described previously in section 6.10.

1. Incorporate a new data source into REA PDB – This includes the process for incorporating a new data source into the REA database.

2. REA table design process – This includes the process for developing/refining the REA database structure based on the source data and other sources such as the Marine Data Model, ISO specifications, etc.

3. REA Lineage processing steps – This includes the process for populating the lineage model as part of REA data management activities.

4. Data extraction workflow for trials – This includes the process for isolating, extracting, and transforming REA data for missions.

Each of these business processes will now be outlined. These processes are illustrated through UML diagramming techniques. The processes are not described to the full extent required for implementation, but rather are outlined to highlight decisions to be made during the process.

8.1 Incorporate a new data source into REA PDB

Incorporating a new data source into the REA PDB is a multi-step process and is illustrated in Figure 17. The existing REAS utilizes separate edit and production structures, both of which may be refined though the process. Ensuring that all data sources within REA PDB are managed and updated appropriately requires a clear understanding of where the data was obtained, how it was integrated within the REA PDB, and when it was last updated.

1. Start: Identify data set to be added.

2. Add data set to inventory spreadsheet: At this time, various aspects of the REA PDB are tracked via a series of spreadsheets. A key component to REA PDB is the data dictionary and source tracking spreadsheet as it includes the current production REA PDB structure, along with the list of sources housed within it.

3. Obtain data set: The data set should be obtained from the originator, along with instructions on receiving updates as needed (may be included in data set metadata).

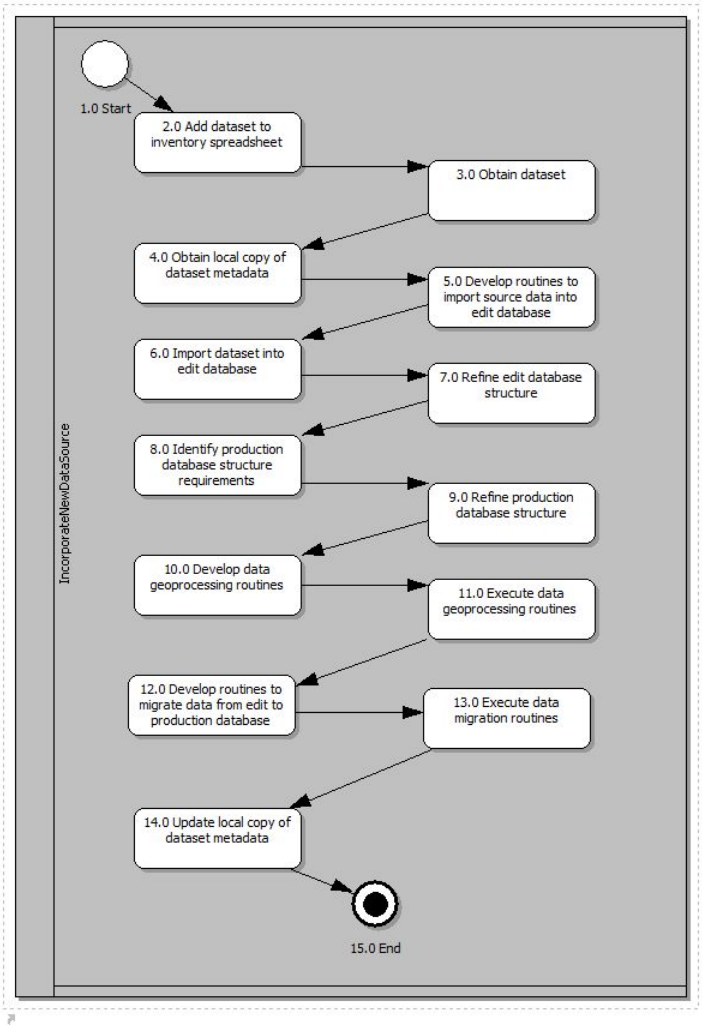

Figure 17: The business process for adding a new data source to the REA PDB.

4. Obtain local copy of data set metadata: When obtaining a copy of a data set managed by a third party, it is important to also obtain a copy of the associated metadata. When the data set is loaded into the load and production databases, the lineage of the data should be maintained for future reference to explain all processing which would have changed or affected the data.

5. Develop routines to import source data into LDB: Ensuring consistent results when reloading a data set requires an established import process with an appropriate level of validation. The effort required and the complexity of the import process will depend upon the source data format.

6. Import data set into LDB: Once the data import routines are in place, the source data set can then be imported into the LDB as required.

7. Refine LDB structure: The LDB structure has been based upon a simple 'dumping ground' model where the complete source data set is imported with no structural modifications. This approach is straight forward with a small number of data sets, but has the disadvantage of requiring individual scripts to load data to production. Developing a managed LDB structure could be beneficial in the long term.

8. Identify production database structure requirements: Incorporating a new data set into the REA LDB requires mapping the data set into the existing REA PDB structure, then possibly refining and extending the REA PDB structure to address new requirements.

9. Refine production database structure: Although refinement of the production database structure would not be a regular occurrence, the review and possible refinement should be considered when incorporating new data sets into the database. The incorporation of a new type of data, as opposed to a new data set of an already registered type, would spawn a more detailed database analysis and design process.

10. Develop data geoprocessing routines: Once the source data set is in the LDB, additional geoprocessing may be required in order for the data set to be compatible with the final production database. Issues such as applying unit conversions, spatial adjustments, and quality flags may be necessary to ensure the usefulness and interoperability of the data.

11. Execute data geoprocessing routines: Any conversions applied to the source data sets should include a follow up validation process.

12. Develop routines to migrate data from load to production database: Each source data set will require a migration process to move from the load to production databases. Where possible, attempts should be made to reduce the number of migration procedures as they must be maintained for future updates to the underlying data sources they support.

13. Execute data migration routines: The migration of data sets from the load to the REA PDB should include a follow up validation process. Once a data source is incorporated into the REA PDB and combined with other sources, errors or other quality issues may be difficult to isolate.

14. Update local copy of data set metadata: Once the data set is loaded into the REA PDB, the 'last uploaded date' attribute of the data set should be updated as well as the local copy of the data set metadata.

15. End

8.2 REA table design process

The table design process (Figure 18) provides an overview of the key resources used in developing the table structures that support new data types and their relative importance.

1. Start:

2. Develop draft table structures based on source data: When integrating a new data set into the REA database, it is important to start by documenting the existing table structures. Initial draft table structures are usually based upon the structure of the source data. The model is then refined based upon existing standards, then by operational business requirements.

3. Refine table structures using Arc Marine data model based entities: The final table structures should be based upon basic marine data types and feature classes within Arc Marine. Unlike standards bodies such as the Open Geospatial Consortium (2009) (OGC) and the ISO, Arc Marine is a more concise resource which can be applied directly. These other standards tend to be more complex and require in depth understanding of their geospatial and other core data type implementations.

4. Refine table structures using OGC based entities: Although the geospatial aspect of the REA database is closer to the OGC than ISO, it is important to be aware that there are areas of overlap between OGC and ISO, and that the OGC standards are based in part on ISO standards. A key difference between the OGC and ISO standards is that the OGC standards can be downloaded freely, whereas the ISO standards must be purchased.

5. Refine table structures using ISO based entities: The International Organization for Standardization is a resource for standards both within and outside the geospatial realm. ISO Technical Committee ISO/TC 211 (2008) is in charge of the development and maintenance of geospatial related standards.

6. Refine table structures based on standard relational database design approaches: By this point, the geospatial aspects of the data model should be established. The development of business tables would be based in part on Arc Marine (includes geospatial and business tables) and the available data sources being incorporated into the REA PDB.

7. Develop and/or link controlled vocabulary lists to table columns where applicable: Where possible, code tables should be based upon an existing controlled vocabulary. Any implemented code lists should be added to the Excel spreadsheet for future reference.

8. End

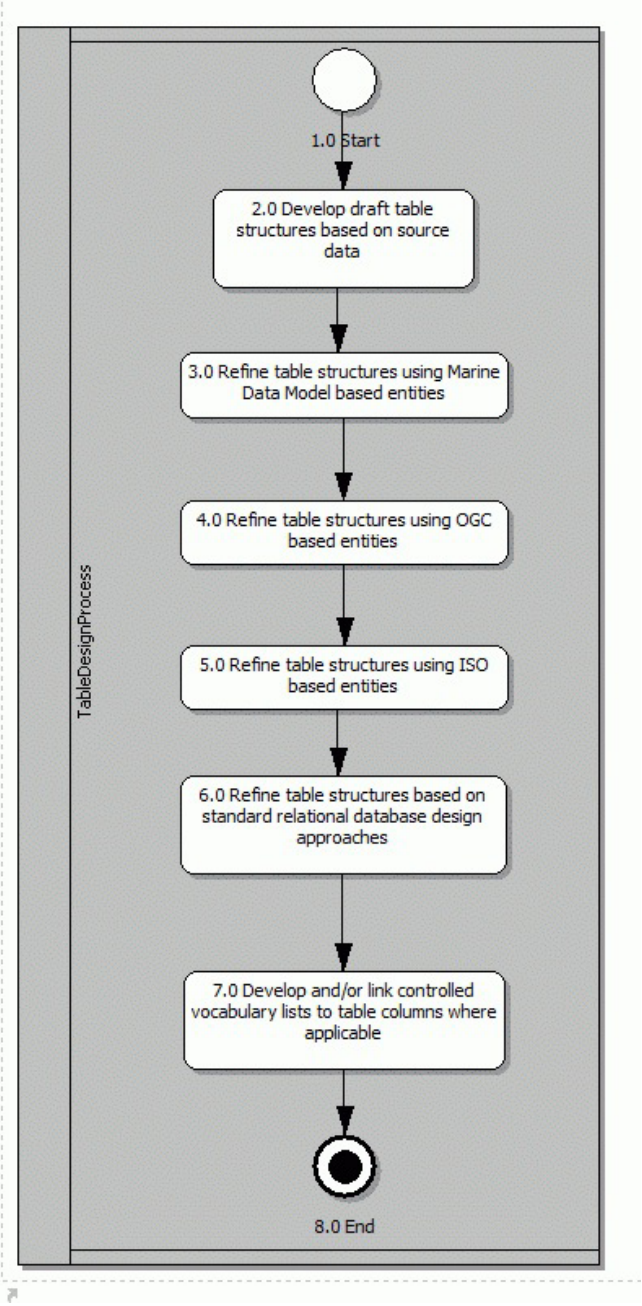

Figure 18: The business process for adding a new table structure to the REA PDB.

8.3 REA lineage processing steps

The lineage processing steps are described below and illustrated in Figure 19. The purpose of the lineage data model is to track the evolution of the described data asset in the REA PDB. Lineage becomes more important with increased number of dependencies (i.e., linkages to externally managed resources) as each externally managed resource is handled differently. The intent of the REA lineage model is to ensure the management of the REA database is tracked, along with updates/additions to the data assets/products from which it is comprised.

1.0 Start:

2.0 Create/update lineage entry: All managed data assets should have a lineage entry created, along with a statement of the lineage of the asset. This is the minimum lineage content required and in the context of the REA PDB, this may not be sufficient given that the PDB will house data from various sources which will be periodically updated.

3.0 Create/update lineage sources: Each source which comprises a part of the REA PDB should be identified and described in sufficient detail to ensure its origin and purpose are clearly understood. The extents of the source (spatial, vertical, temporal, etc), reference system represented by its coordinates, and map scale (where appropriate) should be identified where known and applicable.

3.1 Start

3.2 Create/update lineage source entry: The minimum information requested for a lineage source is the description and map scale (if the source has been prepared for use at a specific scale). In the case of the REA PDB, this level of information may not be sufficient as sources are updated on a periodic basis. Additional details regarding the origin of the data set, how it was obtained, and details of processing required in order to integrate it into the REA PDB are needed to ensure that ongoing updates are managed.

3.3 Create/update lineage source extents: The extent of a lineage source can be defined in many ways: vertical extents, horizontal extents, temporal extents, or descriptive extents (i.e., anecdotal descriptions). In addition, each extent type can be represented in many ways (i.e., instantaneous point, bounding boxes, bounding polygons are examples of horizontal extents). For the purposes of the REA PDB, the primary extent representations will be focused on bounding boxes for horizontal extents, date/time ranges for temporal extents, and minimum/maximum ranges for vertical extents.

3.4 Create/update lineage source citations: All lineage sources should be cited. The citation describes the source, its origin, and provides references to supporting documentation.

3.5 Create/update lineage source contacts: Contacts (responsible parties) should be defined for lineage sources where possible. Contacts may be defined as an organization or individual, and may include supporting details such as contact instructions, addresses, phone/facsimile/e-mail, and associated on-line resources such as web sites and applications.

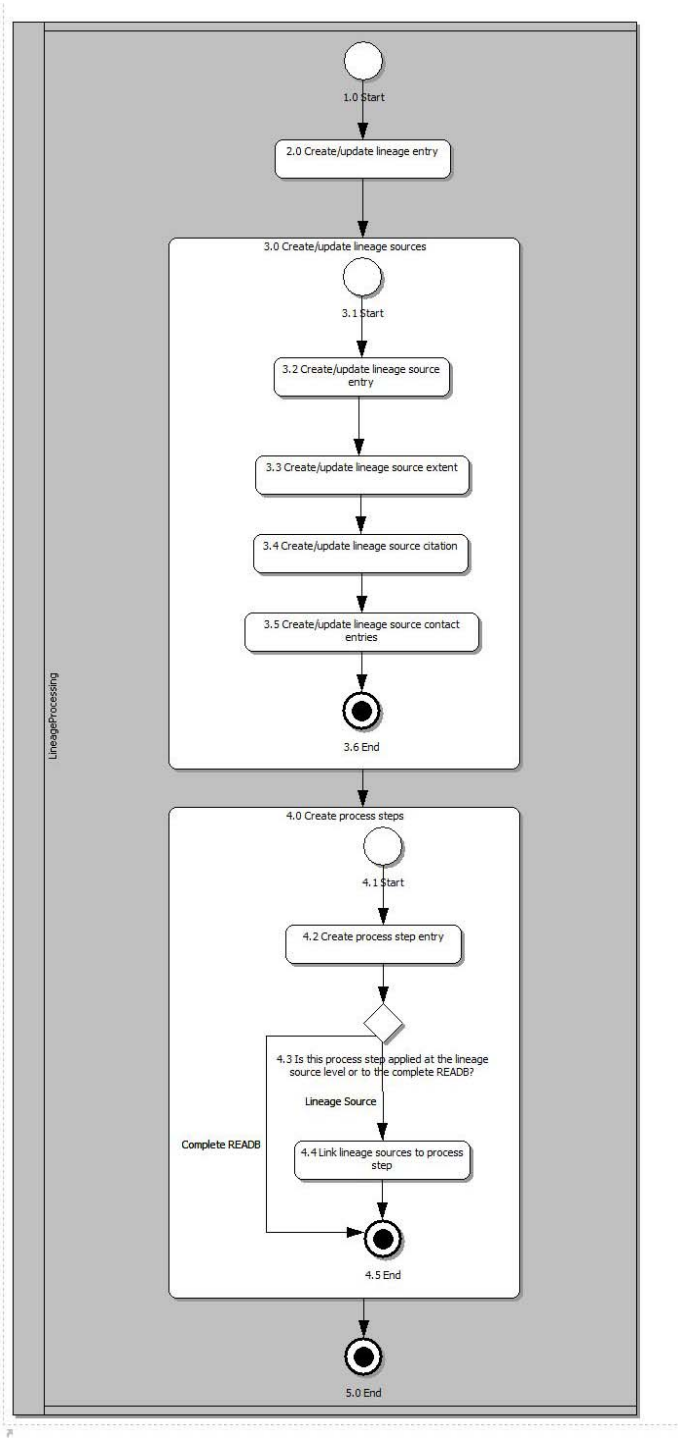

Figure 19: The lineage processing steps.

3.6 End

4.0 Create process steps: Unlike lineage source information, process steps are only created, not updated. They represent the log of processing activity against the REA PDB and its underlying sources. Process steps should follow a consistent format to allow their contents to be parsed and queried for useful information.

4.1 Start

4.2 Create process step entry: Process steps are a running record of changes to an asset at the collection (REA PDB) or source level (lineage source data set). The description attribute of process steps are populated based on the format in the "LineageProcessStepsDescList" list. Note that the semi-colon character is reserved for separating the sections of the process step description.

4.3 Is this process step applied at the lineage source level or to the complete REA database? Source level process steps are created using the same methodology as collection level process steps. The difference is that the process steps are linked to individual sources whereas collection level changes are applied to all sources.

4.4 Link lineage sources to process steps: Within the REA PDB, process steps applied to individual lineage sources are linked to those sources via a join table.

4.5 End

5.0 End

8.4 Data extraction workflow for trials

The primary purpose of the data extraction process is to enable REA PDB users to extract subsets of the database for use in trial scenarios (Figure 20). This approach is more efficient than attempting to deploy and use the entire REA PDB, especially when trials are focused on specific geographic areas.

1.0 Start

2.0 Identify parameters of interest: The first step to extracting a subset of data from the REA PDB is to identify the specific data types required.

3.0 Define extents of interest: Data extracted from the REA PDB is primarily based on physical and temporal extents.

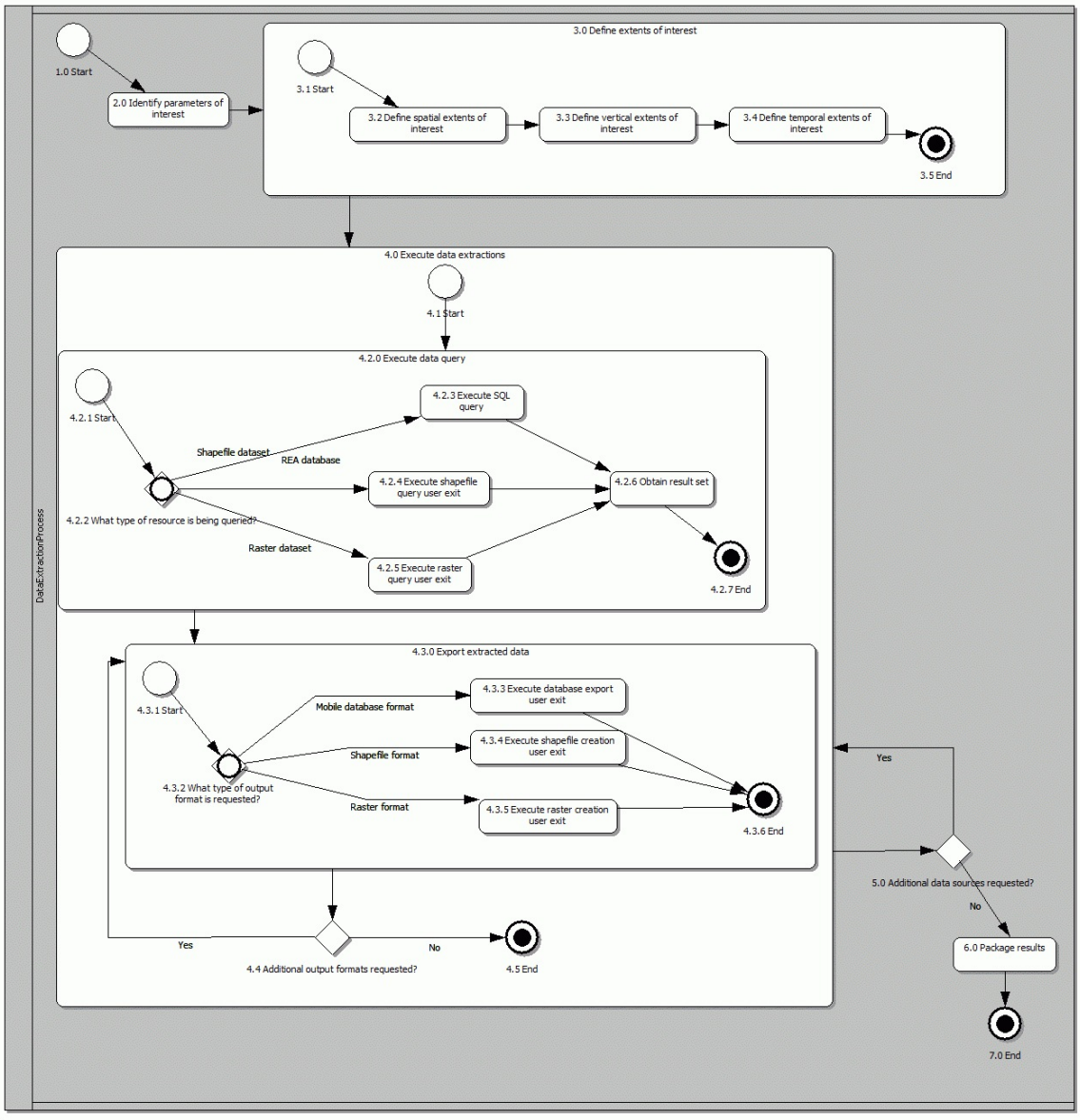

Figure 20: The data extraction workflow for trials.

3.1 Start

3.2 Define spatial extents of interest: The simplest approach to defining spatial extents is through the use of a bounding box. It is possible to define spatial extents using more sophisticated geometric shapes (polygons, rings, etc), which is useful for reducing the amount of data returned and ensuring the data covers only the area of interest (can be important when performing data analysis).

3.3 Define vertical extents of interest: The simplest approach to defining a vertical extent is through the use of a single range. As with spatial extents, vertical extents can also be defined using more sophisticated approaches (multiple ranges), which is useful for reducing the amount of data returned and ensuring the data covers only the area of interest (can be important when performing data analysis).

3.4 Define temporal extents of interest: The simplest approach to defining a temporal extent is through the use of a single range. As with vertical and spatial extents, temporal extents can also be defined using more sophisticated approaches (multiple ranges), which is useful for reducing the amount of data returned and ensuring the data covers only the area of interest (can be important when performing data analysis).

3.5 End

4.0 Execute data extractions: Depending on the number and types of parameters requested, data extraction from the REA PDB involves one or more query and export operations.

4.1 Start

4.2.0 Execute data query: The first step to data extraction is identifying the data records to be exported.

4.2.1 Start

4.2.2 What type of resource is being queried?: The method used for identifying the data of interest will depend on the type of data source being queried.

4.2.3 Execute SQL query: The most common data source being queried is the REA database itself. In this case, standard SQL would be used in identifying the data of interest. PostgreSQL includes common SQL functionality, along with geospatial query extensions.

4.2.4 Execute shapefile query user exit: Shapefiles can be queried using standard SQL or via a user exit. A user exit which supports the spatial column of a shapefile provides functionality above that achieved through a standard SQL query.

4.2.5 Execute raster query user exit: Support for raster queries is highly variable and is heavily dependent on the format of the source. Binary grids can be queried for actual data values whereas raster images can be queried for RGB (red, blue, green) values which may or may not be mapped to specific numerical

values. In the context of the REA PDB, raster queries are executed through a user exit.

4.2.6 Obtain result set: Upon completion of the query process, the data of interest will be obtained/prepared for export.

4.2.7 End

4.3.0 Export extracted data: The second step to data extraction is exporting the data records of interest to required formats.

4.3.1 Start

4.3.2 What type of format is requested?: The output format selected will determine how the data are exported, but also the level of functionality available when using the data after they are exported.

4.3.3 Execute database export user exit: Exporting data in a database format facilitates query and analysis while on the mission.

4.3.6 Execute shapefile creation user exit: Shapefiles are a form of a database table. This format can be advantageous over a database format from a performance and overhead perspective. Shapefiles provide access to data (as opposed to rasters which generally do not) and require little to no infrastructure due to the broad support from free and commercial geospatial APIs.

4.3.5 Execute raster creation user exit: Rasters generally do not retain actual data values. However, they are particularly useful in visualizing data. For example, a high resolution raster of a bathymetric data set, coupled with a generic color map (maps RGB values to approximate data values) can in some cases be sufficient for presenting the meaning of the underlying data set. Using a raster to present such data sets can be much faster than attempting to present the source data.

4.3.6 End

4.4 Additional output formats requested?: If additional output formats are requested, repeat the export operation using the next format.

4.5 End

5.0 Additional data sources requested?: If additional data sources have been requested, repeat the query and export process using the next data source.

6.0 Package results

7.0 End

9 Concluding remarks

This effort has resulted in several specific outputs, including:

- a complete examination and documenting of the existing REA LDB,
- a data model for the next generation REA PDB, and
- the concepts required to construct a REA system.

Each of these represents a tangible output from this effort. In more general terms, we consider other aspects of this work to be useful for other similar database development efforts.

- Data modelling is important

  This effort also highlights the benefits of a well-documented data model for any database development. The act of designing a database is not something that should be left to chance. Unfortunately, the community that utilizes much of the data is unaware of the benefits provided by data modelling and in many cases simply having the data housed within a database appears to meet immediate requirements. As practitioners of data modelling, we must do more to promote the benefits of data modelling and allow managers to make informed decisions about implementation of projects with or without data modelling. If proper design is followed, the process will end with a maintainable product that is understood by everyday users.

- Arc Marine is a valid framework

  In terms of Arc Marine, we consider this framework to be a viable approach for the development of geospatially enabled databases in the oceanographic topic area. The concepts from which Arc Marine was developed help reduce the complexity of a database by allowing the user to manage similar data categories in a similar way. In turn, this contributes to the usability of the database by the users.

  In other communities, database design suffers from the multiple design problems - the creation of multiple models of similar yet different structure. Specialized community groups then form to promote and justify their particular model as the better implementation. Typically, this does not serve interoperability. Arc Marine provides a framework from which a specialized model may be constructed. As such, Arc Marine is not a complete data model but rather the basis from which a specialized model may be constructed. It is hoped that data models following this framework will have sufficient aspects in common so as to promote interoperability.

- The concept of the REA system

  The concept of the REA database was formed out of the DRDC Atlantic desire to have the ability to rapidly assess the ocean environment in which sea trials and missions take place. To accommodate the "rapid" aspect of the requirement, it was realized that better management of our data collections was required. To manage these data, it was decided to develop a database of our collection. This database soon became a location for the storage of whatever data was deemed potentially useful to at-sea activities.

  Storing all potentially useful data within the REA database was eventually recognized as infeasible. From this realization emerged the concept of the REA data system, where the database acted as one component of something larger. This REA system makes use of concepts such as user exits, which contribute to the functionality of the system by enabling the use of data stored both inside and outside REA database. This is a scalable solution to the data volume problem and, with proper management of the data outside the database, it is also a very maintainable solution from the aspect of data product updates.

  The management of data assets contributing to such a system does have an associated cost. Managing a compilation of data assets in such a system can only be reliably achieved through clear business processes and supporting documentation such as package metadata and lineage. Understanding the makeup of the REA database (i.e., its underlying data packages), and the state of the database (i.e., the current state of the various sources), adds to the reliability and usability of the system as a whole.

- Documentation is important

  The redesign of the REA database has been a time consuming and tedious effort. In large part, the complexities of the effort stemmed from our initial lack of understanding of the data model for the existing database. The project spent considerable time understanding and documenting the existing LDB structure. This was required to support any mapping of data from the existing structure to the new structure.

  Although time consuming, the mapping exercise provides an essential output of the project. The mapping provides a pathway for data from the existing database to the redesigned database. It is essential that such a mapping be conducted to ensure the data presently existing in the database have a known and understood position within the redesigned database. The documentation that results from the mapping is the outcome that data administrators can then utilize to make the port from one database to another.

  Documentation in the form of this report is also a critical outcome from this effort. Most users will not delve into the database at such depths to understand all aspects of the tables, fields and relationships. This documentation provides sufficient detail to allow a precursory view of the structure and data contained within it.

- Cost of implementing the data model

  This effort has resulted in a data model for next version of the REA database. This data model has little resemblance to the previous REA database and therefore represents a substantive departure from the existing database.

  As a result, management will need to examine and consider the costs associated with transitioning to the new data model. There will be two primary costs in this transition. First, the costs associated with the effort to port or move the data from one database to another. Second, will be the cost associated with any software modification required as a result of the port. At present we have software that allows interaction with the existing database structure (e.g., PASTET as described by Giles, *et al.* (2009) or the browser interface as described by Deveau (2008)). Obviously, if we wish to maintain this functionality under the new database structure, the software will need to be modified to accommodate the new structure.

  Depending on user requirements, the justification for browser interface modification may be difficult to make. Free GIS tools such as uDig offer the user full GIS capabilities with many of the functions presently available in the developed browser software. With a minor amount of user training, users could be utilizing a full GIS rather than the present software suite developed in-house.

References

Adams, B.D., Bruyn, L.E., Houde, S. and Angelopoulos, P. (2003), Trust In Automated Systems Literature Review, (DRDC Toronto No. CR-2003-096) Defence Research and Development Canada – Toronto.

Buckley, D.J., The GIS Primer: An Introduction to Geographic Information Systems (online), http://www.innovativegis.com/basis/primer/primer.html (Access date: March 10, 2009).

Chapman, D.M.F. and Ellis, D.D. (1998), The Elusive Decibel: Thoughts on Sonars and Marine Mammels, *Canadian Acoustics* 26 (2), 29-31

Chapman, D.M.F., Ellis, D.D. and Staal, P.R. (1997), Assessing the Bottom-interacting Sonar Environment, From the SACLANTCEN Conference.

Computer Associates, Data modelling from CA (online), http://www.ca.com/us/data-modeling.aspx (Access date: March 18, 2009).

Craymer, M. (2006), Making Sense of Evolving Datums: WGS84 and NAD83, From the Hydroscan 2006, Saskatoon, Canada.

Craymer, M.R. (2006), The Evolution of NAD83 in Canada, *Geomatica*, 60 (2), 151-164.

Department of National Defence (1992), Canadian Forces Handbook of Marine Firing Practice and Exercise Areas, (CFTO number A AG H01 003/AG 001).

Deveau, T. (2008), Rapid environmental Assessment Database Project - Phase III, (DRDC Atlantic CR 2008-044) Defence R&D Canada.

Deveau, T.J. (2006), Rapid Environmental Assessment Database Project - Phase I, (DRDC Atlantic CR 2006-213) Defence R&D Canada - Atlantic.

Digital Equipment Corporation (1992), Information Technology - Database Language SQL, ISO 9075:1992.

Divins, D.L., NGDC Total Sediment Thickness of the World's Oceans & Marginal Seas (online), http://www.ngdc.noaa.gov/mgg/sedthick/sedthick.html (Access date: July 24, 2006).

English, L., Ten mistakes to avoid if your data warehouse is to deliver quality information (online), http://dssresources.com/papers/features/english08112002.html (Access date: July 21, 2008).

English, L., Could they build the Sears Tower without a BluePrint? Why Data Modeling is Imperative (online), http://www.dciexpo.com/speakers/archive/english.htm (Access date: July 21, 2008).

ESRI, ESRI - The GIS Software Leader (online), ESRI, http://www.esri.com/ (Access date: March 9, 2009).

Franklin, J.B. (1997), Ambient Noise Measurements on the Canadian Continential Shelf, (DREA CR/97/411) Defence Research Establishment Atlantic.

Gareau, P.L. (2005), GSCA-DRDC Scotian Shelf Geoclutter Project Digital GeoSpatial Data Compilation, (unpublished).

GeoConnections, Mapping the future together online (online), http://cgdi.gc.ca/Welcome.do (Access date: March 10, 2009).

Giles, P., Kilistoff, S. and Brooke, G. (2009), Portable Acoustic Sensitivity Transmission and Estimation Tool, (DRDC Atlantic CR 2008-299) Defence R&D Canada - Atlantic.

GNU, GNU GENERAL PUBLIC LICENSE (online), http://www.gnu.org/copyleft/gpl.html (Access date: March 18, 2009).

GNU, GNU Lesser General Public License (online), http://www.gnu.org/copyleft/lesser.html (Access date: March 11, 2009).

Guptill, D.W. (1994), Documenting, Reorganizing, and Assessing DREA Software for Analysis of Non-Acoustic Data, (DREA CR/94/462) Defence Research Establishment Atlantic.

Hallock, Z.R. and Teague, W.J. (1992), The Fall Rate of the T-7 XBT, *Journal of Atmospheric and Oceanic Technology*, 9, 470-483.

Hanawa, K., Rual, P., Bailey, R., Sy, A. and Szabados, M. (1995), A new depth-time equation for Sippican or TSK T-7, T-6 and T-4 expendable bathythermographs (XBT), *Deep-Sea Research*, 42 (8), 1423-1451.

Hazen, M.G. and Desharnais, F. (1997), The Eastern Canada Shallow Water Ambient Noise Experiment, From the Oceans '97, OCEANS '97. MTS/IEEE Conference Proceedings.

Hutt, D., Osler, J. and Ellis, D. (2002), Effect of hurricane Michael on the underwater acoustic environment of the Scotian Shelf, In *Proceedings of Impact of Littoral Environmental Variability of Acoustic Predictions and Sonar Performance*, SACLANT Undersea Research Centre, Lerici, Italy.

ICES, International Council for Exploration of the Sea (online), http://www.ices.dk (Access date: March 18, 2009).

International Association of Oil and Gas Producers (2006), Surveying and Positioning Guidance note 5; Coordinate reference system definition - recommended practice.

Isenor, A.W. (2008), Enhancing the utility of the Rapid Environmental Assessment database through the use of in situ and modelled data sets during Q316, (DRDC Atlantic TM 2008-212) Defence R&D Canada - Atlantic.

Isenor, A.W. and Lapinski, A.-L.S. (2007), Thoughts on a Design Framework for System Integration, (DRDC Atlantic TM 2006-143) Defence Research and Development Canada.

ISO (2003), Geographic information - Metadata, (ISO 19115:2003(E)) International Organization for Standardization.

ISO/TC 211, Geographic Information/Geomatics (online), http://www.isotc211.org/ (Access date: March 10, 2009).

Junkins, D. and Garrard, G. (1998), Demystifying Reference Systems; A Chronicle of Spatial Reference Systems in Canada, From the SDI'98 Conference, Ottawa.

Kezele, D.B. and Friesen, G. (1993), XBT Test Data Comparison and Analysis, *Sea Technology*, (February), 15-22.

Lauesen, S. and Vinter, O. (2000), Preventing Requirements Defects, From the Proceedings of the Sixth International Workshop on Requirements (REFSQ'2000), Stockholm.

Leung, F. and Bolloju, N. (2005), Analyzing the Quality of Domain Models Developed by Novice Systems Analysts, From the Proceedings of the 38th Hawaii International Conference on System Sciences, IEEE, Hawaii.

Longard, L.R. (1993), Knots, Volts and Decibels, Defence Research Establishment Atlantic.

Mackay, A.G., Hunter, J.A., Hood, R.L. and Chapman, D.M.F. (1986), A 12-Channel Marine Eel for Shallow Refraction Surveying of the Seabottom in Coastal Waters, In T. Akal and J.M. Berkson (Eds.), *Ocean Seismo-Acoustics*, Plenum Publishing Corporation.

Moody, D.L. and Shanks, G.G. (1994), What makes a good data model? Evaluating the quality of entity-relationship models, From the 13th International Conference on the Entity-Relationship Approach, Manchester, UK.

Moody, D.L. and Shanks, G.G. (2003), Improving the quality of data models: empirical validation of a quality management framework, *Information Systems*, 28, 619-650.

Moody, D.L., Sindre, G., Brasethvik, T. and Solvberg, A. (2003), Evaluating the Quality of Information Models: Empirical Testing of a Conceptual Model Quality Framework, From the Proceedings of the 25th International Conference on Software Engineering IEEE Computer Society

National Marine Electronics Association (2002), NMEA 0183 Standard for Interfacing Marine Electronics Devices.

OGP Geodesy Subcommittee (2008), EPSG Geodetic Parameter Dataset, International Association of Oil and Gas Producers.

Open Geospatial Consortium, OpenGIS® Standards and Specifications (online), http://www.opengeospatial.org/standards (Access date: March 10, 2009).

Oregon State University, Arc Marine The ArcGIS Marine Data Model (online), http://dusk.geo.orst.edu/djl/arcgis/index.html (Access date: March 10, 2009).

Osler, J.C., Hines, P.C. and Trevorrow, M.V. (2002), Acoustic and *in-situ* techniques for measuring the spatial variability of seabed geoacoustic parameters in littoral environments, In *Proceedings of Impact of Littoral Environmental Variability of Acoustic Predictions and Sonar Performance*, SACLANT Undersea Research Centre, Lerici, Italy.

Pascal, F. (2000), Practical Issues in Data Management, Addison-Wesley.

PostGIS, PostGIS: Home (online), http://postgis.refractions.net/ (Access date: March 11, 2009).

Ramsey, P. PostGIS Manual, (unpublished).

Refractions Research, uDig User-friendly Desktop Internet GIS (online), http://udig.refractions.net/ (Access date: March 10, 2009).

Seidelmann, P.K. (Ed.) (1992), Explanatory Supplement to the Astronomical Almanac, University Science Books.

Simsion, G. (2007), Data Modeling Theory and Practice, New Jersey: Technics Publications, LLC.

Sparx Systems, Modeling and Design Tools for your Enterprise (online), http://www.sparxsystems.com.au/ (Access date: March 18, 2009).

Standish Group (1995), The CHAOS Report, The Standish Group International.

Stocks, K., Graybeal, J., Neiswender, C. and Isenor, A., (January 21, 2009), Metadata Standards vs. Metadata Specifications (online), Marine Metadata Interoperability Project, http://marinemetadata.org/guides/mdatastandards/stdvsspec (Access date: March 9, 2009).

Stocks, K.I., Neiswender, C., Isenor, A.W., Graybeal, J., Galbraith, N., Montgomery, E.T., Alexander, P., Watson, S., Bermudez, L., Gale, A. and Hogrefe, K., The MMI Guides: Navigating the World of Marine Metadata (online), Marine Metadata Interoperability Project, http://marinemetadata.org/guides (Access date: March 18, 2009).

Whitehouse, B.G. (2004), Survey of ASW Research & Development Opportunities in Rapid Environmental Assessment, (DRDC Atlantic CR 2004-000) Defence R&D Canada - Atlantic.

Wikipedia, Latitude (online), http://en.wikipedia.org/wiki/Latitude (Access date: March 18, 2009).

Wikipedia, Likert scale (online), Wikipedia, http://en.wikipedia.org/wiki/Likert_scale (Access date: March 18, 2009).

Wikipedia, Third normal form (online), http://en.wikipedia.org/wiki/Third_normal_form (Access date: March 18, 2009).

Wright, D., Blongewicz, M.J., Halpin, P.N. and Breman, J. (2007), Arc Marine GIS for a Blue Planet, Redlands, California: ESRI Press.

Wycove Systems Limited (1985), A Data Base for Shallow Water Acoustics, (CR/85/414) Defence Research Establishment Atlantic.

Wycove Systems Limited (1986), SWDB: A Data Base for Shallow Water Acoustics Volume I: User Guide, (CR/86/433) Defence Research Establishment Atlantic.

Annex A Field Mapping from Load to Production Database

The following pages represent the results from the detail mapping exercise. In the following, the source table name is identified for the existing REA DB version 3 beta. Each column name in that table is then detailed, with information including the column type and description. The description is based solely on the investigations in this work. The column E in the table identifies whether or not the data in the source column should be stored in the production database.

Columns F and G refer to the production database as designed in this work. Column F identifies the table in the production database where the source data should be stored. Column G identifies the column name in the indicated production database table.

Legend:

Green highlighted box: LDB field data is mapped to a location in the PDB.

Rose highlighted box: Table is accounted for using User Exits. See section 6.10.

	A	B	C	D	E	F	G
1	**Source Table**					**Production Table**	
2	**Table Name**	**Column Name**	**Column type**	**Column Definition**	**Keep in PDB (Yes/No)**	**Production Table Name**	**Production Column Name**
3	**_atable**	id_filename			NO		
4		id			NO		
5		atext			NO		
6							
7	**temp_nadas_lines_decoded_parts**	id			NO		
8		id_filename			NO		
9		drdc_code			NO		
10		thetime			NO		
11		thetimestamp			NO		
12		p1			NO		
13		p2			NO		
14		p3			NO		
15		p4			NO		
16		p5			NO		
17		p6			NO		
18		p7			NO		
19		p8			NO		
20							
21	**temp_view_xbt_file_meta_data**	id			NO		
22		filename			NO		
23		id_cruise			NO		
24		header_bottom_depth_m			NO		
25		header_bucket_c			NO		
26		header_bucket_depth			NO		
27		header_cruise			NO		
28		header_date_of_launch_mdy			NO		
29		header_depth_coefficient_1			NO		
30		header_depth_coefficient_2			NO		
31		header_depth_coefficient_3			NO		
32		header_depth_equation			NO		
33		header_depth_m			NO		
34		header_display_units			NO		
35		header_operator			NO		
36		header_pressure_point_correcti			NO		
37		header_pressure_point_correcti			NO		
38		header_pressure_point_correcti			NO		
39		header_probe_type			NO		
40		header_raw_data_filename			NO		
41		header_salinity_ppt			NO		
42		header_sequence_number			NO		
43		header_serial			NO		
44		header_ship			NO		
45		header_surface_temp_c			NO		
46		header_terminal_depth_m			NO		
47		header_time_of_launch_hms			NO		
48		header_water_depth_m			NO		
49		header_water_temp_c			NO		
50		header_latitude_d_dm_h			NO		
51		header_longitude_d_dm_h			NO		
52							
53	**atlbathy**	id	integer		USER EXIT		
54		depth	real		USER EXIT		
55		the_geom	geometry		USER EXIT		
56							
57	**authorization_table**	toid	oid	The numeric representation of the similarly named column.	NO		

	A	B	C	D	E	F	G
58		rid	text	The numeric representation of the similarly named column.	NO		
59		expires	timestamp	Empty field.	NO		
60		authid	text	Empty field.	NO		
61							
62	**bathy_filenames**	id			USER EXIT		
63		filename			USER EXIT		
64		filedate			USER EXIT		
65		chars			USER EXIT		
66		lines			USER EXIT		
67		maxlen			USER EXIT		
68							
69	**bathy_lines**	id			USER EXIT		
70		depth			USER EXIT		
71		id_filename			USER EXIT		
72		the_geom			USER EXIT		
73							
74	**bellhop_data**	id		Unique identifier.	NO		
75		id_bellhop_q		The id used in the bellhop_q table.	NO		
76		range		The range of the prediction.	NO		
77		bearing		The bearing direction of the individual radials.	NO		
78		the_geom		The geometry for the latitude and longitude position of the model run.	NO		
79							
80	**bellhop_plots**	id		Unique field.	NO		
81		the_geom		The geometry field.	NO		
82		id_bellhop_q		The id used in the bellhop_q table.	NO		
83							
84	**bellhop_q**	id		Unique identifier.	NO		
85		rxdepth		The user defined depth of the receiver.	NO		
86		nradials		The number of radials being requested in the calculation.	NO		
87		dbthreshold			NO		
88		shortd		A short user description which identifies the model run.	NO		
89		the_geom			NO		
90		the_plot			NO		
91							
92	**bodc_biota_comp_model**	code			NO		
93		param			NO		
94		sub_name			NO		
95		sub_altname			NO		
96		taxon_code			NO		
97		taxon_name			NO		
98		taxon_class			NO		

	A	B	C	D	E	F	G
99		param_comp			NO		
100		comp			NO		
101		comp_class			NO		
102		samp_prep			NO		
103		analysis			NO		
104		data_proc			NO		
105		created			NO		
106		modified			NO		
107							
108	**bodc_category**	code			NO		
109		title			NO		
110		docref			NO		
111		record_clock			NO		
112		created			NO		
113		modified			NO		
114							
115	**bodc_category_link**	category_code			NO		
116		group_code			NO		
117							
118	**bodc_chem_model**	code			NO		
119		param			NO		
120		param_cl			NO		
121		sub_name			NO		
122		sub_name_cl			NO		
123		sub_altname			NO		
124		param_comp			NO		
125		comp			NO		
126		comp_class			NO		
127		comp_phase			NO		
128		comp_phase_cl			NO		
129		samp_rep			NO		
130		analysis			NO		
131		data_proc			NO		
132		created			NO		
133		modified			NO		
134							
135	**bodc_itis_map**	code			NO		
136		param			NO		
137		taxon_code			NO		
138		taxon_name			NO		
139		taxon_class			NO		
140		param_comp			NO		
141		comp			NO		
142		comp_class			NO		
143		samp_prep			NO		
144		analysis			NO		
145		data_proc			NO		
146		created			NO		
147		modified			NO		
148		bioentrf			NO		
149							
150	**bodc_parameter**	code			NO		
151		group_code			NO		
152		unit_code			NO		
153		dummy_val			NO		
154		min_permiss_val			NO		
155		max_permiss_val			NO		
156		before_dp			NO		
157		after_dp			NO		
158		sig_fig			NO		
159		short_title			NO		

	A	B	C	D	E	F	G
160		full_title			NO		
161		definition			NO		
162		record_lock			NO		
163		bodc_legal			NO		
164		created			NO		
165		modified			NO		
166							
167	**bodc_parameter_group**	code			NO		
168		bodc_legal			NO		
169		created			NO		
170		definition			NO		
171		full_title			NO		
172		modified			NO		
173		record_lock			NO		
174		short_title			NO		
175							
176	**bodc_units**	code			NO		
177		short_title			NO		
178		full_title			NO		
179		comments			NO		
180		record_lock			NO		
181		created			NO		
182		modified			NO		
183							
184	**cities**	gid	integer	Unique identifier with values from 0-24.	NO		
185		name	character varying	The name of the city or community. These are all in Canada.	NO		
186		capital	character varying	A code value of N,Y,C indicating no, yes, country capital, respectively.	NO		
187		prov_name	character varying	The province within which the community exists.	NO		
188		population	character varying	Population estimate of the community.	NO		
189		the_geom	geometry	The geometry XY position.	NO		
190							
191	**codesources**	id	integer	Unique identifier for the records.	NO		
192		longdescription	character varying	Contains 3 records; short descriptions of "DRDC NADAS"; "DRDC SWDB"; "DRDC XBT"	NO		
193		shortdescription	character varying	Longer description continaing "Phase I - initial data loading"	NO		
194							
195	**country**	gid	integer	Unique identifier.	NO		
196		area	real	Area value in unknown units.	NO		
197		perimeter	real	Perimeter in unknown units.	NO		
198		cntry_	integer	Unknown	NO		
199		cntry_id	integer	Unknown	NO		

	A	B	C	D	E	F	G
200		fips_cntry	character varying	Two character country code.	NO		
201		gmi_cntry	character varying	Three character country code.	NO		
202		cntry_name	character varying	Country name.	NO		
203		sovereign	character varying	Country name.	NO		
204		pop_cntry	real	Population at an unknown time.	NO		
205		curr_type	character varying	Type of monatary currency in use.	NO		
206		curr_code	character varying	The code that indicates the type of currency Eg CAD, USD.	NO		
207		landlocked	character varying	Indicates whether or not the country is landlocked. Both values are N.	NO		
208		color_map	character varying	Unknown	NO		
209		sovereign_	character varying	Who is sovereign nation residing over the land mass.	NO		
210		the_geom	geometry	geometry	NO		
211							
212	**cruises**	id	integer	Primary Key. This is the cruise number.	YES	Cruise Ships_On_Cruise Scientist_On_Cruise Instantaneous_Point	Cruise_ID = fieldcontent
213		longdescription	character varying	Long description of the cruise. Mostly a blank field.	YES	Cruise_Notes	Cruise_ID = cruises.id Counter = sequential counter foe each cruiseID Note = fieldcontent
214		shortdescription	character varying	Short description of the cruise. Mostly a blank field but also contains the cruise number (eg. Q234)	NO		
215		id_chief_scientist	integer	ID number of the chief scientist (if identified - there are lots of blank entries).	YES	Scientist_On_Cruise	Scientist_ID = sequentialnumber Scientist_Name = name from Scientist table
216		id_chief_scientist_2	integer	ID number of the 2nd chief scientist (if identified - again lots of blank entries).	YES	Scientist_On_Cruise	Scientist_ID = sequentialnumber Scientist_Name = name from Scientist table
217							
218	**data_columns**	column_name	text	Name of XBT metadata column. Only 6 names in the table.	NO		
219		id_bodc_code	integer	zero content.	NO		
220		id_drdctypes	integer	zero content.	NO		
221		id_isocode	integer	zero content.	NO		
222		table_name	text	All 6 values contain xbt_file_meta_data	NO		
223		id	integer	Numeric value 1 to 6.	NO		
224		id_filetype	integer	Always value of "2".	NO		

	A	B	C	D	E	F	G
225							
226	**directory_specifications**	id	integer	Primary key.	NO		
227		directory_specification	character varying	The directory from which the load file originated. Only 3 directories are given in his table.	NO		
228							
229	**drdc_units**	id		Contain a single value of 1 record.	NO		
230		longdescription		Contains the single record with "no units defined"	NO		
231		notes		Contains the text "catch all for any code without units"	NO		
232		shortdescription		Contains the text "no units"	NO		
233							
234	**drdctypes**	id		Unique identifier which is also the same numeric as the NADAS record line identifiers. SOME of these numbers should be moved to the Parameter table - but not all. This table is full of extraneous information on things like Event Marks - which really are not data parameter names.	YES	Parameter	Parameter_ID
235		id_bodc_code		Unclear what this is. All fields are empty.	NO		
236		id_codesource		This seems to be a 1 for NADAS codes, a 2 for XBT codes and 3 for Shallow water DB codes.	NO		
237		id_drdc_units		All records empty.	NO		
238		id_drdctype_parent		This is used to nest the parameter codeing into parents and children.	NO		
239		id_isocode		Empty.	NO		
240		is_trustedcode		This is a t or f value indicating true or false. But I don't know what is being judged for trustworthyness.	NO		
241		longdescription		Long description but doesn't look useful for explaining the parameter.	NO		
242		shortdescription		Short description of the parameter. Parts of this could be used in our redesign.	YES	Parameter	Description
243							
244	**ecs_flights**	id	double precision	Primary key. This can be used for flight number.	NO		
245		date	timestamp with time zone	Date stamp (no time) for each flight.	YES	Instantaneous_Point	Date_Time

	A	B	C	D	E	F	G
246		date2	integer	A day counter from some unknown zero point. Values are like 33000.	NO		
247		windsite1	real	Wind value in knots measured at site 1	YES	Data	Parameter_ID needs to recognize that as WIND ESTIMATE Device_ID should indicate Proxy (ie Sea State used to estimate wind) Data_Value = fieldcontent
248		windsite2	real	Wind value in knots measured at site 2	YES	Data	Parameter_ID needs to recognize that as WIND ESTIMATE Device_ID should indicate Proxy (ie Sea State used to estimate wind) Data_Value = fieldcontent
249		windsite3	real	Wind value in knots measured at site 3	YES	Data	Parameter_ID needs to recognize that as WIND ESTIMATE Device_ID should indicate Proxy (ie Sea State used to estimate wind) Data_Value = fieldcontent
250		windsite4	real	Wind value in knots measured at site 4	YES	Data	Parameter_ID needs to recognize that as WIND ESTIMATE Device_ID should indicate Proxy (ie Sea State used to estimate wind) Data_Value = fieldcontent
251		wind	real	This field does not have wind values in it. I'm note sure what these values are. We can trace the values back to the spreadsheet, but that doesn't enlighten us. This field should be ignored.	NO		
252							
253	**ecs_sites**	id	integer	Primary key. This is actually the Site number.	YES	Instantaneous_Point	Station_ID
254		the_geom	geometry	Point geometry	YES	Measurement_Location	Geom = fieldcontent Z could be 30 m depth if the_geom does not have the z value stored internally
255		location	text	A general name for the location of the site.	YES	Ambient_Noise	Location_Comment = fieldcontent
256		sediment	text	A general comment on the type of sediment at the site.	YES	Ambient_Noise	Sediment_Comment = fieldcontent
257		depth_m	real	An approximate depth at the site.	YES	Ambient_Noise	Approximate_Depth = fieldcontent
258		shipping	text	A general comment on the shipping activity at the site.	YES	Ambient_Noise	Shipping_Comment = fieldcontent
259							
260	**ecs_wind_obs**	id	integer	Unique id for the records in the table.	NO		
261		id_flight	integer	The Flight number. This field contains 1, 2, 3, or 4.	YES	Instantaneous_Point	Survey_ID

	A	B	C	D	E	F	G
262		id_site	integer	The site number. There were 4 sites, numbered 1,2,3,4. The site id matches the id in ecs_sites.	YES	Instantaneous_Point	Station_ID but this must take into account the three apex positions for each site. Thus, we will end up with 12 Station_ID values (4 sites, 3 apices).
263		hz	real	The frequency at which the ambient noise data was obtained.	YES	Data	Parameter_ID = something suitable for frequency DeviceID for all records will be something suitable for SONOBUOY. Data_Value = fieldvalue
264		north_mean	real	The 1 hour temporal mean at the north apex of a site indicated in field id_site.	YES	Data	Note different location Parameter_ID = something suitable for ambient noise mean Data_Value = fieldvalue
265		north_stdev	real	The temporal standard deviation of the ambient noise obtained from the northern apex.	YES	Data	Parameter_ID = something suitable for ambient noise standard deviation Data_Value = fieldvalue
266		east_mean	real	The 1 hour temporal mean at the east apex of a site indicated in field id_site.	YES	Data	Note different location Parameter_ID = something suitable for ambient noise mean Data_Value = fieldvalue
267		east_stdev	real	The temporal standard deviation of the ambient noise obtained from the eastern apex.	YES	Data	Parameter_ID = something suitable for ambient noise standard deviation Data_Value = fieldvalue
268		west_mean	real	The 1 hour temporal mean at the west apex of a site indicated in field id_site.	YES	Data	Note different location Parameter_ID = something suitable for ambient noise mean Data_Value = fieldvalue
269		west_stdev	real	The temporal standard deviation of the ambient noise obtained from the western apex.	YES	Data	Parameter_ID = something suitable for ambient noise standard deviation Data_Value = fieldvalue
270		m_m	real	The mean of the mean values in all three directions.	NO		
271		sd_m	real	In most cases, this is the standard deviation of the three mean values. However, the initial data load was not conducted properly and in some cases this field contains a mean value of previous measurements between 100 hz and 1995 hz. When this occurs, the hz value is zero.	NO		
272		m_sd	real	The mean of the three standard deviation values.	NO		
273		sd_sd	real	The standard deviation of the three standard deviation values.	NO		
274							
275	**ecs_wind_obs_vs_model**	id	integer	Unique id for the records in the table.	NO		

	A	B	C	D	E	F	G
276		id_site	integer	The site number. There were 4 sites, numbered 1,2,3,4. The site id matches the id in ecs_sites.	NO		
277		id_flight	integer	The Flight number. This field contains 1, 2, 3, or 4.	NO		
278		knots	real	The measured wind speed at the particular site and flight numbers.	NO		
279		obs500hz	real	The observed noise at this wind speed, site, flight and frequency combination.	NO		
280		obs800hz	real	The observed noise at this wind speed, site, flight and frequency combination.	NO		
281		obs1260hz	real	The observed noise at this wind speed, site, flight and frequency combination.	NO		
282		obs2000hz	real	The observed noise at this wind speed, site, flight and frequency combination.	NO		
283		model500hz	real	The modelled noise at this wind speed, and frequency combination.	NO		
284		model800hz	real	The modelled noise at this wind speed, and frequency combination.	NO		
285		model1260hz	real	The modelled noise at this wind speed, and frequency combination.	NO		
286		model2000hz	real	The modelled noise at this wind speed, and frequency combination.	NO		
287							
288	**etopo2**	id			USER EXIT		
289		the_geom			USER EXIT		
290		depth			USER EXIT		
291							
292	**etopo5**	id			USER EXIT		
293		depth			USER EXIT		
294		the_geom			USER EXIT		
295							
296	**filenames**	id	integer	Primary key. Some files appear to have been loaded in the NADAS observations. For example, NADAS observations indicates an id_filename =607. However, no such id exists in this table for the id field. As well, id_filename 607 does not exist in geopoints either.	NO		
297		filename	character varying	This is the name of the file used in the load.	NO		

	A	B	C	D	E	F	G
298		id_cruise	integer	The cruise number. Since this is an integer, this cannot handle characters such as A, B. The A, B designation is sometimes used to address multiple "legs" of a cruise.	NO		
299		id_directory_specification	integer	This is a sequential identifier for the directory from which the load file originated.	NO		
300		id_filename_parent	integer	I am not sure what this is. But all records in the table have a value 0 for this field.	NO		
301		id_filetype	integer	A sequential file type identifier. Eg 2 indicates an XBT profile file.	NO		
302		id_scientist	integer	I suspect it was suppose to be a sequential identifier for a scientist. However, this field is blank throughout the table.	NO		
303		id_ship	integer	A sequential ship identifier number eg. 1 indicates Quest.	NO		
304		nlines	integer	The number of lines in the input load file.	NO		
305		nmaxlinelength	integer	The column where the last character is located in the line with the maximum width in the input load file.	NO		
306							
307	**filetypes**	id	integer	Primary key; A generated sequential counter for the type of file. At present, only 4 identifiers exist.	NO		
308		longdescription	character varying	A long description for the type of load file.	NO		
309		shortdescription	character varying	A short description for the type of load file.	NO		
310							
311	**gc_airgun_profile_sections**	gid			USER EXIT		
312		objectid			USER EXIT		
313		layer			USER EXIT		
314		shape_leng			USER EXIT		
315		profile_no			USER EXIT		
316		the_geom			USER EXIT		
317							
318	**gc_bedforms_basinatlas**	gid			USER EXIT		
319		attibutes			USER EXIT		
320		the_geom			USER EXIT		
321							
322	**gc_bedforms_frm_sidescan**	attibutes			USER EXIT		
323		the_geom			USER EXIT		
324		gid			USER EXIT		
325							
326	**gc_bedrock_geology**	gid			USER EXIT		
327		area			USER EXIT		

	A	B	C	D	E	F	G
328		perimeter			USER EXIT		
329		code			USER EXIT		
330		formation			USER EXIT		
331		the_geom			USER EXIT		
332							
333	**gc_bedrock_outcrops**	gid			USER EXIT		
334		objectid			USER EXIT		
335		layer			USER EXIT		
336		the_geom			USER EXIT		
337							
338	**gc_bouyancy_line_moraines**	gid			USER EXIT		
339		objectid			USER EXIT		
340		layer			USER EXIT		
341		shape_leng			USER EXIT		
342		the_geom			USER EXIT		
343							
344	**gc_drift_outcrops**	gid			USER EXIT		
345		area			USER EXIT		
346		perimeter			USER EXIT		
347		layer			USER EXIT		
348		the_geom			USER EXIT		
349							
350	**gc_ed_nav_of1427**	gid			USER EXIT		
351		objectid			USER EXIT		
352		area			USER EXIT		
353		perimeter			USER EXIT		
354		ed_nav_of1			USER EXIT		
355		ed_nav_o_1			USER EXIT		
356		cruise_no			USER EXIT		
357		daytime			USER EXIT		
358		modifier			USER EXIT		
359		thickness			USER EXIT		
360		label			USER EXIT		
361		angle			USER EXIT		
362		the_geom			USER EXIT		
363							
364	**gc_epicentres**	gid			USER EXIT		
365		objectid			USER EXIT		
366		layer			USER EXIT		
367		the_geom			USER EXIT		
368							
369	**gc_faults**	gid			USER EXIT		
370		objectid			USER EXIT		
371		layer			USER EXIT		
372		shape_leng			USER EXIT		
373		the_geom			USER EXIT		
374							
375	**gc_fishtrawl**	gid			USER EXIT		
376		objectid			USER EXIT		
377		layer			USER EXIT		
378		shape_leng			USER EXIT		
379		the_geom			USER EXIT		
380							
381	**gc_groundfish**	gid			USER EXIT		
382		objectid			USER EXIT		
383		layer			USER EXIT		
384		shape_leng			USER EXIT		
385		shape_area			USER EXIT		
386		the_geom			USER EXIT		
387							
388	**gc_gsca_ship_tracks**	gid			USER EXIT		

	A	B	C	D	E	F	G
389		objectid			USER EXIT		
390		shape_leng			USER EXIT		
391		cruise_no			USER EXIT		
392		the_geom			USER EXIT		
393							
394	**gc_iceberg_furrows**	gid			USER EXIT		
395		objectid			USER EXIT		
396		layer			USER EXIT		
397		shape_leng			USER EXIT		
398		shape_area			USER EXIT		
399		the_geom			USER EXIT		
400							
401	**gc_infilled_channels**	gid			USER EXIT		
402		objectid			USER EXIT		
403		layer			USER EXIT		
404		shape_leng			USER EXIT		
405		the_geom			USER EXIT		
406							
407	**gc_isopach_contour**	gid			USER EXIT		
408		tnode_			USER EXIT		
409		length			USER EXIT		
410		contour_id			USER EXIT		
411		dxf_color			USER EXIT		
412		line_type			USER EXIT		
413		thicklabel			USER EXIT		
414		lineconfid			USER EXIT		
415		the_geom			USER EXIT		
416							
417	**gc_isopach_thickness**	gid			USER EXIT		
418		objectid			USER EXIT		
419		layer			USER EXIT		
420		shape_leng			USER EXIT		
421		shape_area			USER EXIT		
422		the_geom			USER EXIT		
423							
424	**gc_nearshore_bedrock**	gid			USER EXIT		
425		objectid			USER EXIT		
426		layer			USER EXIT		
427		shape_leng			USER EXIT		
428		shape_area			USER EXIT		
429		the_geom			USER EXIT		
430							
431	**gc_nongeoclutter_surveysitespoly**	gid			USER EXIT		
432		objectid			USER EXIT		
433		shape_leng			USER EXIT		
434		shape_area			USER EXIT		
435		name			USER EXIT		
436		the_geom			USER EXIT		
437							
438	**gc_pockmarks**	gid			USER EXIT		
439		objectid			USER EXIT		
440		layer			USER EXIT		
441		the_geom			USER EXIT		
442							
443	**gc_rib_moraines**	gid			USER EXIT		
444		objectid			USER EXIT		
445		layer			USER EXIT		
446		the_geom			USER EXIT		
447							
448	**gc_sand_ridges_chs**	gid			USER EXIT		
449		entity			USER EXIT		

	A	B	C	D	E	F	G
450		layer			USER EXIT		
451		elevation			USER EXIT		
452		thickness			USER EXIT		
453		color			USER EXIT		
454		bedform_id			USER EXIT		
455		grainsize			USER EXIT		
456		the_geom			USER EXIT		
457							
458	**gc_scotian_shelf_regional_surfical_geology**	gid			USER EXIT		
459		area			USER EXIT		
460		perimeter			USER EXIT		
461		geo_sym			USER EXIT		
462		code			USER EXIT		
463		formation			USER EXIT		
464		texture			USER EXIT		
465		dataset_id			USER EXIT		
466		spatial_id			USER EXIT		
467		the_geom			USER EXIT		
468							
469	**gc_seabed_texture_frm_sidescan**	gid			USER EXIT		
470		attibutes			USER EXIT		
471		the_geom			USER EXIT		
472							
473	**gc_till_tongues**	gid			USER EXIT		
474		objectid			USER EXIT		
475		layer			USER EXIT		
476		direction			USER EXIT		
477		the_geom			USER EXIT		
478							
479	**geoareas**	gid	integer	Unique identifier for the records. This table is identical to table gc_scotian_shelf_regional_surfical_geology.	NO		
480		area	character varying	The spatial polygon area for each polygon in the multipolygon. Units unknown.	NO		
481		perimeter	character varying	The perimeter of each polygon in the multipolygon. Units unknown.	NO		
482		geo_sym	integer	A total of 7 unique values exist for this field: 3; 5; 8; 100; 224; 326; 402. I'm not sure what these indicate.	NO		
483		code	character varying	A total of 7 different codes: 6; 7; 8; 9; 10a; 10b; 10c. I don't know what these are.	NO		
484		formation	character varying	A text description that identifies a particular type of bottom. Examples include "Sambro Sand", "Emerald Silt", "Lahave Clay".	NO		
485		texture	character varying	A total of 7 descriptions of the bottom.	NO		
486		dataset_id	character varying	Empty.	NO		

	A	B	C	D	E	F	G
487		spatial_id	character varying	Empty.	NO		
488		the_geom	the geometry; MULTIPOLYGON.	Geometry - MULTIPOLYGON. The reference system used appears to be metres. But when these data are placed on a map, they collect at the North Pole. not sure what is going on here.	NO		
489		id_drdctype	integer	Note different attribute name as compared to "data_columns" table (ie. no s). The only content is "900" which does not appear in the drdctypes table as an id number.	NO		
490							
491	**geocircles**	gid	integer	Empty field.	NO		
492		quality	integer	Empty field.	NO		
493		gid_geopoint	integer	Empty field.	NO		
494		id_filename	integer	Empty field.	NO		
495		fileline	integer	Empty field.	NO		
496		time	timestamp	Empty field.	NO		
497		id_drdctype	integer	Empty field.	NO		
498		radius	real	Empty field.	NO		
499		the_geom	geometry	Empty field.	NO		
500							
501	**geolines**	gid	integer	A unique ID for the line.	NO		
502		quality	integer	All records contain 0 (i.e. zero)	NO		
503		gid_geopoint1	integer	A gid identifier that relates back to a single gid in the geopoints table. This is the first point in the line.	NO		
504		gid_geopoint2	integer	A gid identifier that relates back to a single gid in the geopoints table. This is the second point in the line.	NO		
505		the_geom	linestring	The geometry representing the line.	NO		
506							
507	**geometry_columns**	f_table_catalog	character varying 256	Empty field.	NO		
508		f_table_schema	character varying 256	Value of public for all valid record.	NO		
509		f_table_name	character varying 256	The name of the table in the READB	NO		
510		f_geometry_column	character varying 256	The name of the column in the above named table.	NO		

	A	B	C	D	E	F	G
511		coord_dimension	integer	The dimension of the coordinate. Value of 2 for all valid records.	NO		
512		srid	integer	The srid number corresponding to the coordinate system used in the geometry.	NO		
513		type	character varying 30	The geometry type. POINT: LINESTRING: MULTILINESTRING: POLYGON: MULTIPOLYGON	NO		
514							
515	**geopoints**	gid	integer	Primary Key; generated sequential counter	NO		
516		id_filename	integer	Foreign key link to filenames identifier. These ID numbers are grouped in the sense that all numbers > 1 million and < 2 million appear to apply to XBT data.	NO		
517		fileline	integer	This is the line number from the input load ASCII file for the XBT profile. If I examine an XBT profile record in geopoints, and for one record I get the fileline number, trace this back to the id column in z1_xbt_lines table with id=fileline, it always indicates a LATITUDE input line from the LOAD ASCII file. I see absolutly no reason to have this information in this table.	NO		
518		time	timestamp	Date and time associated with the point. For the XBT profiles, the time in this field appears to be the correct launch time. For NADAS records, it is the time from the 013 coded strings.	YES	Instantaneous_Point	Date_Time = fieldcontent
519		quality	integer	Was intended to be the quality indicator for something. Presently contains only values 1(51 records) and 0 (2.5 million records). This content is of no use.	NO		
520		id_drdctype	integer	I am not sure what this is for. There are only two different values in this field - the value 603 (51 records) and the value 0 (2.5 million records). This field is not Foreign keyed to the table drdctypes; nor is the content really applicable as the 603 drdctype id is for 'Latitude' and the 0 value has no corresponding record in drdctypes.	NO		
521		the_geom	geometry	Point geometry string.	YES	Measurement_Location	Geom = fieldcontent; but individual z values will have to be added to the Geom for each individual point.
522							
523	**gom15dd**	id	integer		USER EXIT		
524		depth	real		USER EXIT		
525		the_geom	geometry		USER EXIT		
526							

	A	B	C	D	E	F	G
527	**isocodes**	id	integer	No content.	NO		
528		isocode	character varying	No content.	NO		
529							
530	**logins**	id	text	Unique sequential number for the user.	YES	Logins	id = fieldcontent
531		password	text	Character password for the user.	YES	Logins	password = fieldcontent
532		username	integer	Usename for the user.	YES	Logins	username = fieldcontent
533							
534	**marloa**	gid	integer	A unique gid identifier, but not the same gid as in geoareas.	NO		
535		id	integer	A unique idenitifier.	NO		
536		label	text	Abbreviated version of the sea area name. Eg. N3.	NO		
537		sea_areas	text	The name of the area, in full spelling. Eg. November Three	Yes	Feature_Area	Feature_Code=fieldcontent
538		air_space	text	Only two values present; "To 20,000 feet" and "To 30,000 feet"	NO		
539		employment	text	Only 5 rows with content. Looks like a text description the subsurface area.	NO		
540		file_name	text	All values contain "Firing Practice Area"	NO		
541		common_nam	text	All values contain "FP&EX_Areas"	NO		
542		projection	text	All values contain "not projected"	NO		
543		geo_area	text	All values contain "East Coast of Canada"	NO		
544		source	text	All values contain the same long string of text referring to Notice to Mariners.	NO		
545		creator	text	All values contain "Hydrographic Services Office (Atlantic)"	YES	Feature_Characteristic	Value=fieldcontent; while Name gets the original field name.
546		department	text	The department. All values contain Department of National Defence.	YES	Feature_Characteristic	Value=fieldcontent; while Name gets the original field name.
547		country	text	All values contain "Canada"	YES	Feature_Characteristic	Value=fieldcontent; while Name gets the original field name.
548		date	text	All values contain "2002May05".	YES	Feature_Characteristic	Value=fieldcontent; while Name gets the original field name.
549		the_geom	geometry	the geometry; MULTIPOLYGON.	YES	Feature_Area	Geom=fieldcontent
550							
551	**nadas_codes**	id	integer	The NADAS numeic code that is found on each NADAS record line.	NO		
552		id_codesource	integer	Unknown. Contains all ones (ie 1).	NO		
553		is_trustedcode	boolean	A boolean t or f. Not sure what it actually means or applies to.	NO		

	A	B	C	D	E	F	G
554		longdescription	character varying	Mostly blank, but sometimes contains a longer description of the numeric code.	NO		
555		shortdescription	character varying	The short description of what the code numeric means.	NO		
556							
557	**nadas_file_meta_data**	id	integer	some type of numeric counter. There are presently 1262 distinct values in this field. Yet the field goes to a maximum of 1335. I don't know why these are not sequential.	NO		
558		header_local_offset	integer	contains only zero values	NO		
559		header_present_julian_day	integer	contains only zero values	NO		
560		header_present_time	timestamp without time zone	This is not the first time in the file; it is not the time corresponding to code 013; I don't know what this is.	NO		
561		header_raw_filename	character varying	Empty for all rows.	NO		
562		id_filename	integer	An ID that can be used with filenames.id to identify the original file. Can also be used with geopoints id_filename.	NO		
563		lat_max	double precision	Appears to be the maximum latitude recorded in the ASCII nadas file for code 013.	NO		
564		lat_min	double precision	I assume this is the minimum latitude value in the ASCII input file.	NO		
565		lon_max	double precision	I assume this is the maximum longitude value in the ASCII input file.	NO		
566		lon_min	double precision	I assume this is the minimum longitude value in the ASCII input file.	NO		
567		lat_start	double precision	Appears to be the first latitude in the ASCII nadas file with code 013.	NO		
568		lon_start	double precision	Appears to be the first longitude in the ASCII nadas file with code 013.	NO		
569		lat_end	double precision	Appears to be the last latitude in the ASCII nadas file with code 013.	NO		
570		lon_end	double precision	Appears to be the last longitude in the ASCII nadas file with code 013.	NO		
571							
572	**nadas_observations**	id	interger	Unique numeric identifier.	NO		

	A	B	C	D	E	F	G
573		air_temperature_degc	real	The air temperature - unknown source.	YES	Data	Data_Value = fieldcontent Parameter_ID = 55 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
574		barometric_pressure_mbar	real	The barometric presure - unknown source.	YES	Data	Data_Value = fieldcontent Parameter_ID = 52 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
575		course_over_ground_degt	real	ship course over ground	YES	Data	Data_Value = fieldcontent Parameter_ID =135 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
576		depth_m	real	This is a sounding value, NOT depth of measurement.	YES	Data	Data_Value = fieldcontent Parameter_ID =308 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
577		gid_geopoint	interger	Perhaps the link back to the geopoints table.	NO		
578		percipitation_mm	real	Multiple values may be present in the original data stream. These should be averaged for insertion here.	YES	Data	Data_Value = fieldcontent Parameter_ID = 53 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
579		relative_humidity_pct	real	Multiple values may be present in the original data stream. These should be averaged for insertion here.	YES	Data	Data_Value = fieldcontent Parameter_ID = 51 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
580		ship_heading_degt	real	the direction the ships bow is pointing	YES	Data	Data_Value = fieldcontent Parameter_ID = 127 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
581		ship_propeller_port_rpm	real	rotational frequency of port side propeller	YES	Data	Data_Value = fieldcontent Parameter_ID = 130 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
582		speed_over_ground_kt	real	the speed the ship is making relative to ground	YES	Data	Data_Value = fieldcontent Parameter_ID = 134 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.

	A	B	C	D	E	F	G
583		surface_water_temperature_degc	real	Surface water temperature - unknown source.	YES	Data	Data_Value = fieldcontent Parameter_ID = 50 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
584		time	timestamp	Time and date. As far as I can tell, this date and time is the only column with date/time information related to the NADAS data. Based on id_filename=607, I cannot locate in geopoints any points for these data.	YES	Instantaneous_Point	Date_Time = fieldcontent
585		wind_speed_apparent_kt	real	Apparent wind speed in knots.	NO		
586		id_filename	real	The identifier associated with the input filename.	NO		
587		wind_speed_kt	real	The computer wind speed in degrees true based on combining both anemometer measurements and taking into account ship speed and direction. NOTE: there is a subtle timing error in the NADAS output that results in an inconsistent temporal reporting of the wind data in the NADAS data stream. The data from the individual sensors are not at the same instance in time as the end result computed speed and direction. So, you cannot check the computation in the NADAS system, nor can you recreate the computation from the existing data stream. That is because the data provided from the sensors is not the data used in the computation of the end result speed and direction.	YES	Data	Data_Value = fieldcontent Parameter_ID = 128 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
588		wind_dir_degt	real	The computer wind direction in degrees true based on combining both anemometer measurements and taking into account ship speed and direction.	YES	Data	Data_Value = fieldcontent Parameter_ID = 129 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
589		wind_stbd_speed	real	Wind speed from starboard anemometer.	NO		
590		wind_port_speed	real	Wind speed from port anemometer.	NO		
591		wind_port_dir	double precision	Wind direction from port anemometer.	NO		
592		wind_stbd_dir	real	Wind direction from starboard anemometer.	NO		
593		wind_stbd_speed_apparent	real	The apparent wind speed from the starboard side anemometer.	NO		
594		wind_dir_apparent	real	Apparent wind direction.	NO		

	A	B	C	D	E	F	G
595		wind_port_speed_apparent	real	Apparent wind speed based on port side anemometer.	NO		
596		wind_stbd_dir_apparent	real	Apparent wind direction based on starboard side anemometer.	NO		
597		wind_port_dir_apparent	real	Apparent wind direction based on port side anemometer.	NO		
598		macradar	text	Empty in all rows.	NO		
599		tsk	text	Empty in all rows.	NO		
600		visibility_solar	real	Soloar visibility - unknown source.	YES	Data	Data_Value = fieldcontent Parameter_ID = 57 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
601		ship_propeller_stbd_rpm	real	rotational frequency of starboard side propeller	YES	Data	Data_Value = fieldcontent Parameter_ID = 131 Device_ID = it should be possible to track the devices via log books maintained by Bob MacDonald.
602		the_geom	geometry	The geometry field.	NO		
603							
604	**opareas**	id	integer	Primary key. Sequential counter.	NO		
605		sea_areas	text	The name of the sea operation area. Full name, all uppercase.	YES	Feature_Area	Feature_Code = fieldcontent
606		air_space	text	The air space height limit.	YES	Feature_Characteristic	Value = fieldcontent ; Name is assigned "AirSpaceHeightLimit"
607							
608	**opareas_areas**	id	integer	Primary key. Sequential counter.	NO		
609		id_oparea	integer	The primary key "id" counter from the opareas table.	NO		
610		notes	text	Only 3 records there were notes in the coordinates field from the DND publication that defined the areas.	NO		
611		charts	text	The charts that the_geom polygon applies to.	NO		
612		the_geom	geometry	The geometry as defined by POSTGIS system.	NO		
613		bounds	boolean	Contains text values "t" or "f", indicating true or false. I suspect this indicates whether or not there is complete overlap of the chart and the op area.	NO		
614							
615	**opareas_points**	id	integer	Primary key. Sequential counter.	NO		
616		id_opareas_area	integer	The primary key "id" counter from the opareas_areas table.	NO		

	A	B	C	D	E	F	G
617		the_geom	geometry	These are the point values that define the corners of the opareas.	NO		
618							
619	**scientists**	scientist	text	Scientist first and last name.	YES	Scientist	Scientist_Name = DISTINCT(fieldcontent)
620		id	integer	Primary key. Unique ID for the scientist. NOT sequential.	NO		
621							
622	**scientists_vs_filename**	id_filename	integer	Unique idenitfier for a file.	NO		
623		scientist	text	The scientist name in all different forms (first initial last name; full first and last name; one space or many spaces)	NO		
624							
625	**sediment_ss_position**	id	integer	Unique identifier.	NO		
626		lat	double precision	Latitude	NO		
627		lon	double precision	Longitude	NO		
628		group_num	integer	Unknown, but is a simple counter identical to the id field.	YES	Data	Parameter_ID = Data_Value = fieldcontent
629		num_measurements	integer	Number of measuremnts that were collected. This may not be the number in the mean velocity calculation.	YES	Data	Parameter_ID = Data_Value = fieldcontent
630		velocity_kps	real	Mean velocity of sound in the bottom (kilometres per second).	YES	Data	Parameter_ID = Data_Value = fieldcontent
631		error_kps	real	Error associated with mean velocity (kilometres per second).	YES	Data	Parameter_ID = Data_Value = fieldcontent
632		high_cluster	integer	The number of points in a cluster which exists more than one standard deviation from the mean, on the side of increased velocity.	YES	Data	Parameter_ID = Data_Value = fieldcontent
633		low_cluster	integer	The number of points in a cluster which exists more than one standard deviation from the mean, on the side of decreased velocity.	YES	Data	Parameter_ID = Data_Value = fieldcontent
634		average_clearance_m	real	The average distance of the measureing equipment from the seabed.	YES	Data	Parameter_ID = Data_Value = fieldcontent
635		the_geom	geometry	Geometry.	YES	Measurement_Location	Geom
636							
637	**sedthick**	id	integer	Unique ID for the records. A total of 9.3 million records in this table.	NO		
638		the_geom	geometry	The x and y position of the sediment data.	YES	Mesh_Point	Geom

	A	B	C	D	E	F	G
639		thickness	real	The sediment thickness in metres. These numbers can be as high as 20,000. I have verified that this is in metres. In ScalarQuantity, the datetime field will have to be filled with something appropriate for the entire sediment data set.	YES	Scalar_Quantity	Data_Value
640							
641	**ships**	id	integer	Primary key.	NO		
642		shipname	character varying	The name of the ship. Currently, only one ship name exists in the table (ie Quest).	YES	Ship	Ship_Name = DISTINCT(fieldcontent)
643							
644	**spatial_ref_sys**	srid	integer	3162 records in this table. This is a POSTGIS system table and should be recreated by the POSTGIS system. We do not provide a mapping because the identical table must be used in the production database.	YES		
645		auth_name	varying character (256)	The only name that exists is EPSG - stands for European Petroleum Survey Group.	YES		
646		auth_srid	integer	This field always equals the srid field.	YES		
647		srtext	varying character (2048)	A text representation of the spatial reference system.	YES		
648		proj4text	varying character (2048)	Information on the projection including projection coefficients.	YES		
649							
650	**swdb_geocircles**	quality	integer	Unknown. All values are set to 1.	NO		
651		gid_geopoint	integer	Possible the gid values in geopoint. It it not clear what these values are.	NO		
652		radius	real	The radius of the circle as determined from the range of the shot.	NO		
653		the_geom	geometry	Circle geometry	NO		
654		id_radius	integer	The numeric id of the specific radius.	NO		
655		id_filename	integer	The id of the filename from which the initial transmission loss data originated.	NO		
656		gid	integer		NO		
657							
658	**swdb_geopoints**	gid	integer	The unique geopoint ID. However, this gid is not contained in geopoints table. the_geom in this table is used as the geomtery when mapping.	NO		
659		id_filename	integer	A unique identifier back to the filename table, for field id.	NO		

	A	B	C	D	E	F	G
660		fileline	integer	The line number of the record as read from the input file.	NO		
661		time	timestamp	A date and time value. But I can't seem to match any of these times to the time values provided in the headers of the raw input files.	NO		
662		quality	integer	Contains a single value of 1.	NO		
663		id_drdctype	integer	Contains a single value of 603. According to table drdctypes, id=603 indicates 'latitude'. Another mystery.	NO		
664		the_geom	geometry	The geometry of the latitude longitude position.	NO		
665							
666	**swdb_index**	id	integer	Unique ID for the records in the table.	NO		
667		id_filename	integer	The filename of the original data file.	NO		
668		gid_geopoint	integer	The geopoint ID used in the geopoint table.	NO		
669		tag_file_name	text	The name of the computer file that holds the data (must not contain dashes).	NO		
670		tag_station_number	text	The station number of the cruise at which the data were collected. 01A,99Z	YES	Instantaneous_Point	Station_ID = fieldcontent
671		tag_area_name	text	The chart or popular name of the area within which the data were collected.	YES	Survey_Info	Description = fieldcontent
672		tag_latitude	text	The latitude of the station (+ is North, - is South). Values -90 to +90. Note that this is the latitude of the hydrophone mooring, and not the position of the shot drops.	YES	Measurement_Location	Geom; For the mooring location, the z value must be obtained from the sta_depth field (station depth field).
673		tag_longitude	text	The longitude of the station (- is West, + is East). Values -180 to +180. Note that this is the latitude of the hydrophone mooring, and not the position of the shot drops.	YES	Measurement_Location	Geom; For the mooring location, the z value must be obtained from the sta_depth field (station depth field).
674		tag_run_number	text	The experimental run number associated with the data collection activity. Multiple runs can exist in a single cruise. Values 00-99.	YES	Survey_Info	Description = fieldcontent
675		tag_st_date	text	The date of the start of the run. 000101,991231	YES	Instantaneous_Point Survey_Info	Date_Time Start_Date_Time
676		tag_st_time	text	The time of the start of the run. 000001,240000	YES	Instantaneous_Point Survey_Info	Date_Time Start_Date_Time
677		tag_en_time	text	The time of the end of the run. 000001,240000	YES	Instantaneous_Point Survey_Info	Date_Time End_Date_Time

	A	B	C	D	E	F	G
678		tag_bearing	text	The bearing (in degrees true) of the run relative to the station.	YES	Propagation_Loss	Bearing = fieldcontent
679		tag_source_depth	text	The depth (m) below the surface of the charges or projectors used.	YES	Propagation_Loss	Source_Depth = fieldcontent
680		tag_sta_depth	text	The water depth (m) at the station (at the array itself).	YES	Propagation_Loss	Station_Depth = fieldcontent Also used in Geom for mooring location.
681		tag_min_depth	text	The minimum water depth (m) along the run.	YES	Propagation_Loss	Minimum_Run_Depth = fieldcontent
682		tag_max_depth	text	The maximum water depth (m) along the run.	YES	Propagation_Loss	Maximum_Run_Depth = fieldcontent
683		tag_hp_depth	text	A list of hydrophone depths.	YES	Device_Class_Detail	Descriptor = hp_depth Value = parsed fieldcontent
684		tag_wired_numbers	text	A list of the serial numbers of the electronic boards in the pots. The data in this field is identical to the data in tag_hp_numbers field for all values in the table.	YES	Device_Class_Detail	Descriptor = hp_number Value = parsed fieldcontent
685		tag_bottom_type	text	A description of the bottom type: mud, clay, sand, gravel, chalk, etc.	YES	Propagation_Loss	Bottom_Type = fieldcontent
686		tag_ss_profile	text	A description of the sound speed profile. isospeed, downward refracting, etc.	YES	Propagation_Loss	Sound_Description = fieldcontent
687		tag_sea_state	text	The international sea state number on a scale of 0-6, which prevailed during the run.	YES	Propagation_Loss	Sea_State = fieldcontent
688		tag_cruise_number	text	The number of the DREA cruise during which the data were collected.	YES	Instantaneous_Point	Cruise_ID = fieldcontent
689		tag_direction	text	The general direction of the run: OPENING,CLOSING,CIRCULAR.	YES	Propagation_Loss	Line_Direction = fieldcontent
690		tag_min_range	text	The miminum range of the source in kilometres.	YES	Propagation_Loss	Minimum_Range = fieldcontent
691		tag_max_range	text	The maximum range of the source in kilometres.	YES	Propagation_Loss	Maximum_Range = fieldcontent
692		tag_hp_numbers	text	A list of the numbers of the actual ceramic hydrophone in the pots. The data in this field is identical to the data in tag_wired_numbers for all values in the table.	NO		
693		tag_bandwidth	text	The bandwidth of the data. 1/3OCTAVE,OCTAVE,NARROWBAND	YES	Propagation_Loss	Band_Width = fieldcontent
694		tag_creation_date	text	The date at which the header and run were loaded. 000101,991231	YES	Propagation_Loss	Creation_Date

	A	B	C	D	E	F	G
695		tag_creation_time	text	The time at which the header and run were loaded. 000001,240000	YES	Propagation_Loss	Creation_Date
696		tag_date_last_edit	text	The date of the last edit to the header and/or run data. 000001,240000	YES	Propagation_Loss	Edit_Date
697		tag_time_last_edit	text	The time of the last edit to the header and/or run data. 000001,240000	YES	Propagation_Loss	Edit_Date
698		tag_no_of_shots	text	The number of shots presented for each hydrophone. 0,99	YES	Propagation_Loss	No_Of_Shots = fieldcontent
699		tag_no_of_frequencies	text	The number of frequencies presented for each hydrophone. 0,99	YES	Propagation_Loss	No_Of_Frequencies = fieldcontent
700		tag_frequency	text	A list of the central frequencies of the data bins.	YES	Data	Parameter_ID = unique number for centre frequency Data_Value = fieldcontent
701		tag_run_number_extension	text	Letter extension to run number taken from PDP file name.	YES	Survey_Info	Description = fieldcontent
702		tag_notes	text	These are the comments contained in record type 4 in the input files.	YES	Profile_Notes	Note = fieldcontent
703		tag_rab_file	text	The name of the range and bearing file for source ship.	NO		
704		tag_xbt_file	text	The name of the file with the bathymetric and temperature data.	NO		
705		tag_noise_file	text	A list of the names of the associated ambient noise files.	NO		
706		tag_hp_position	text	A list of horizontal positions of the hydrophones. This is measured from the knee of the mooring line.	YES	Device_Class_Detail	Descriptor = hp_position Value = parsed fieldcontent
707		id_cruise	integer	The numeric cruise number.	NO		
708		nshots	integer	The numeric number of shots.	NO		
709		nfrequencies	integer	The numeric number of frequencies.	NO		
710		nhydrophones	integer	The numeric number of hydrophones.	NO		
711		tag_ranges	text	A comma separated list of all the ranges at which shots were deployed.	YES	Data	Parameter_ID = unique number for range Data_Value = fieldcontent (but parsing is required)
712		tag_shot_numbers	text	A comma separated list of all the shot numbers.	YES	Data	Parameter_ID = unique number for shot number Data_Value = fieldcontent (but parsing is required)
713		time	timestamp	This time value seems to be offset by 1 hour. If you look at cruise 144, and compare the time in this field with the start time, the time in this field is one hour later. This also applied to cruise 70, with the same 1 hour shift in time.	NO		
714							

	A	B	C	D	E	F	G
715	**swdb_tl_files**	id	integer		NO		
716		fileline	integer		NO		
717		filecontent	character varying	The exact content of this field cannot be stored in Data_Value. The content is a mirror image of the input files. However, this is the only location where the actual propagation loss data exist in the current READB version 3b. So if we are going to use the existing DB as a load source for the production DB, this is the only location from which we can obtain the propagation loss data.	YES	Data	Data_Value = fieldcontent (but with parsing required.
718		gid_geopoint	integer		NO		
719		id_filename	integer		NO		
720		id_shot	integer		NO		
721		id_hydrophone	integer		NO		
722		range	real		NO		
723							
724	**testme**	id	integer	A test file.	NO		
725		mytext	text	A test file.	NO		
726		the_geom	geometry	A test file.	NO		
727		raster	oid	A test file.	NO		
728							
729	**tsd_geopoints**	pid	integer	The position ID. An identifier used to link to the other pid values in the other tsd tables. There are 42076 records in this table, which represents the full grid.	NO		
730		depth	double precision array	An array of 15 elements, which define the vertical depth of the points.	YES	Mesh_Point	Geom
731		the_geom	geometry	The XY position of the grid point.	YES	Mesh_Point	Geom with depth obtained from depth field.
732							
733	**tsd_salinity**	pid	integer	An identifier used to link to the other pid values in the other tsd tables. The pid and the month together are the primary key for the table. The pid values are repeated 12 times,1 for each month.	NO		
734		sal	double precision array	Salinity values in an array of 15 elements. These are the 15 depths defined in tsd_geopoints.	YES	Scalar_Quantity	Data_Value
735		month	integer	Month value 1-12	YES	Scalar_Quantity	Date_Time
736							
737	**tsd_temperature**	pid		An identifier used to link to the other pid values in the other tsd tables. The pid and the month together are the primary key for the table. The pid values are repeated 12 times,1 for each month.	NO		

	A	B	C	D	E	F	G
738		temp		Temperature values in an array of 15 elements. These are the 15 depths defined in tsd_geopoints.	YES	Scalar_Quantity	Data_Value
739		month		Month value 1-12	YES	Scalar_Quantity	Date_Time
740							
741	**type_xref_column_name**	id	integer		NO		
742		column_name	text		NO		
743		id_bodc_code	integer		NO		
744		id_drdctypes	integer		NO		
745		id_isocode	integer		NO		
746		id_table_name	integer		NO		
747							
748	**type_xref_table_name**	id	integer		NO		
749		table_name	text		NO		
750							
751	**userqueries**	longdescription	character varying		NO		
752		shortdescription	character varying		NO		
753		sqlstring	character varying		NO		
754		id	integer		NO		
755		owner	text		NO		
756		vars	text		NO		
757							
758	**wh_depth**	gid			USER EXIT		
759		fnode			USER EXIT		
760		tnode			USER EXIT		
761		lpoly			USER EXIT		
762		rpoly			USER EXIT		
763		length			USER EXIT		
764		gom15ctr			USER EXIT		
765		gom15ctr_i			USER EXIT		
766		contour			USER EXIT		
767		the_geom			USER EXIT		
768							
769	**xbt_file_meta_data**	id	int4	Primary Key.	NO		
770		bucket_temperature_degc	varchar	An independent temperature obtained using a bucket of water and thermometer at the time of XBT launch. A SELECT on the DB shows no bucket temperatures in the DB. Thus, we omit this field.	NO		
771		date_of_launch	varchar	The date of the launch. Since this is a varchar field, and since the initial content is entered at the time of launch, the format of this will NOT be consistent throughout the table.	NO		

	A	B	C	D	E	F	G
772		depth_coefficient_1	varchar	The first coefficient used in the equation that computes depth from time. NOTE: These coefficients appear to be messed up. Typically, Z=at - bt^2, with a and b being the two coefficients. Note there are two coefficients, not three. From what I can tell, there are problems with the +- sign on some coefficients and also the order of the coefficients. I suspect this had no impact on the data, because I would expect errors like this to make depths terribly wrong.	YES	XBT_Profile	Name = CoefficientA or no entry if fieldcontent = zero Value = fieldcontent
773		depth_coefficient_2	varchar	The second coefficient used in the equation that computes depth from time.	YES	XBT_Profile	Name = CoefficientA or B (see *) Value = fieldcontent * If depth_coefficient_1 has a non-zero value, then depth_coefficient_2 becomes CoefficientB. Otherwise it is CoefficientA.
774		depth_coefficient_3	varchar	The third coefficient used in the equation that computes depth from time.	YES	XBT_Profile	Name = CoefficientB or no entry if fieldcontent = zero or EMPTY Value = fieldcontent
775		depth_equation	varchar	One might think this is the exact equation used for the drop rate equation - you would be wrong. The field contains the word "Standard" or a null.	YES, but will be in a different form. The word STANDARD really has no mean by itself.	XBT_Profile	Name = Equation Value = fieldcontent (which is typically 'STANDARD').
776		depth_m	varchar	Sounding depth in metres. May contain trailing "M", or " M', or no indicator, or only "M" (ie no depth value).	YES	XBT_Profile	Sounding = fieldcontent
777		display_units	varchar	Contains only the text "METRIC". I suspect this had something to do with historic display of the values.	NO		
778		gid_geopoint	int4	Field contains all zeros. Perhaps was intended to be a link field to the geopoints table.	NO		
779		id_filename	int4	A unique file identifier used to track the load file name.	NO		
780		probe_type	varchar	The type of probe used for the specific profile. NOTE the probe could be a specific XBT probe, or a SV (sound velocity) probe. You might be wondering why an SV probe is in a table with the name "XBT" - good question. There are only 5 distinct records and of these, only 3 are really distinct devices.	YES	Measuring_Device	Device_ID = {2; 3; 4} Name = {XBTT5; XBTT7; XSV02}

	A	B	C	D	E	F	G
781		raw_data_filename	varchar	The name of the original binary file, from which was created an ASCII file, which was then used as load source for the DB. This name DOES contain a path to the load directory; but it is a path on a PC.	NO		
782		salinity_ppt	varchar	Some type of salinity estimate, probably at the surface, probably near the time of launch. Units will vary between ppt and ppm. Values may have trailing "PPT", "PPM", with or without preceding spaces.	YES	XBT_Profile	Assumed_Salinity = fieldcontent
783		sequence_number	varchar	The drop number on the cruise. It appears that sequence number is equivalent to 'operation number' or 'event number' or 'station number'. I would expect variations to exist between different cruises. In some cases, sequence number may be linked to probe type.	YES	Instantaneous_Point	Station_ID = fieldcontent
784		serial_number	varchar	I would assume it is the serial number of the probe - but I am uncertain. Only 36 unique numbers exist in this field. Most fields contain zeros. Even the fields with valid looking numbers are often duplicated in the rows, leading me to believe these are not the SNs of the XBTs.	YES	XBT_Profile	Serial_Number = fieldcontent
785		surface_temperature_degc	varchar	A surface water temperature that probably originated with the NADAS data - again I am uncertain.	YES	XBT_Profile	Surface_Temperature = fieldcontent
786		terminal_depth_m	varchar	The maximum depth capable from this profile type.	YES	Device_Class_Detail	Descriptor = XBTTerminalDepth Value = fieldcontent There will only be one terminal depth per XBT type.
787		time	timestamp	This is a date and time field. The date appears correct - but I suspect this is conditional on the input format from the text entered date of launch field. The time component is incorrect - it is set to 00:00:00.. However, the file used for load does have a valid time. In other words, it appears that all the launch times have NOT made their way to this DB table. The launch time values are in the geopoints table, in the 'time' field.	NO		
788		x_depth_coefficient_1	float4	The numeric representation of the similarly named column.	NO		
789		x_depth_coefficient_2	float4	The numeric representation of the similarly named column.	NO		

	A	B	C	D	E	F	G
790		x_depth_coefficient_3	float4	The numeric representation of the similarly named column.	NO		
791		x_depth_m	float4	The numeric representation of the similarly named column.	NO		
792		x_salinity_ppt	float4	The numeric representation of the similarly named column.	NO		
793		x_sequence_number	float4	The numeric representation of the similarly named column.	NO		
794		x_serial_number	float4	The numeric representation of the similarly named column.	NO		
795		x_surface_temperature_degc	float4	The numeric representation of the similarly named column.	NO		
796		x_terminal_depth_m	float4	The numeric representation of the similarly named column.	NO		
797		x_bucket_temperature_degc	float4	The numeric representation of the similarly named column.	NO		
798		the_geom	geometry	Point geometry string. This is the position of the XBT profile.	YES	Measurement_Location	Geometry = fieldcontent with z value obtained from depth_m field in xbt_profiles table below.
799							
800	**xbt_profiles**	id	int4	Primary key.	NO		
801		id_filename	int4	A numeric identifier for the file from which the data was loaded. IDs start at 1000001 and increase.	NO		
802		depth_m	float4	The computed depth of the measurement for the profile. Depth is in metres from the sea surface. Depths can be expected below the sounding or bottom depth, because the data have never been qualtiy controlled.	YES	Measurement_Location	Goes to the z component of the Geometry.

	A	B	C	D	E	F	G
803		soundspeed_mps	float4	The computed soundspeed for the specific depth. Sound speed is in metres/sec. Soundspeed can be expected below the sounding or bottom depth because the data have never been quality controlled. I have done SELECT DISTINCT on all tables that contain id_drdctype or id_drdctypes and have not found id=617 to be used. Thus, we will redefine 617 to be "sound speed" and not "sound speed profile". Then we will use 617 for this paramter value. Some data in this table is from sound velocity meters. These can be identified by the presence of a sound velocity value, with no temperature value. So, there are two types of data in this field: one from a device (sound velocity meter) and one computed (based on XBT temperature profile).	YES	Data	For data from a Sound Velocity Meter, the data goes in: Parameter_ID = 617 Device_ID = 3 Data_Value = fieldcontent For data computed from a XBT temperature profile, the data goes in: Parameter_ID = 617 Device_ID = 1 Data_Value = fieldcontent
804		temperature_degc	float4	Measured temperature for the specific depth. Temperature in degrees celsius. Temperatures can be expected below the sounding or bottom depth because the data have never been quality controlled.	YES	Data	Parameter_ID = 322 Device_ID = 2 or 3 depending on xbt_file_meta_data.probe_type Data_Value = fieldcontent
805							
806	**z1_nadas_filenames**	id			NO		
807		numberoflines			NO		
808		maxlinesize			NO		
809		filename			NO		
810		id_cruise			NO		
811		notes		Empty note field.	NO		
812		date_added			NO		
813							
814	**z1_nadas_lines**	id			NO		
815		id_ldseq			NO		
816		id_filename			NO		
817		line			NO		
818							
819	**z1_swdb_filenames**	id			NO		
820		filename			NO		
821		loadname			NO		
822		nlines			NO		
823		nwidth			NO		
824		notes		Only 2 notyes both related to load issues.	NO		
825		is_tl_file			NO		
826							
827	**z1_swdb_lines**	id			NO		
828		id_filename			NO		
829		line			NO		
830							
831	**z1_xbt_filenames**	id			NO		
832		numberoflines			NO		

	A	B	C	D	E	F	G
833		maxlinesize			NO		
834		filename			NO		
835		id_cruise			NO		
836		notes		Empty note field.	NO		
837							
838	**z1_xbt_lines**	id	int		NO		
839		id_filename	int	Same file ID number as in xbt_profile.	NO		
840		line	char(255)	This is an exact copy of the lines read from the original XBT ASCII files.	PARTS	Profile_Notes	Station_ID = xbt_file_meta_data.sequence_number for match of id_filename Note = fieldcontent when content begins with // string.

This page intentionally left blank.

Annex B REA production database table names and comments

Table 6: Complete list of table names and table comments in the PDB.

Table Name	Table Comment/Description
Ambient_Noise	Information on the ambient noise experiment. Data that pertains to the entire experiment.
Area_Characteristic	Contains the name and value of a characteristic that applies to a specified area. Examples of such characteristics include the minimum measured data values for all profiles from the area, of a specific data type.
Mesh_Asset	Table to link the asset number to the mesh ID. By defining the asset ID number, we allow the mesh to be described as an asset, and then define the data content of the mesh as being dependent on that mesh definition.
CI_Citation	A citable reference for or associated with the data set. See ISO 19115 element #359.
CI_Responsible_Party	Identification of a person or organization associated with the data set. See ISO 19115 element #374.
Cruise	The main table to declare a cruise as a data collection activity.
Cruise_Notes	Cruise_Notes contains any notes to be associated with the cruise.
CTD	CTD table contains metadata specific to the CTD cast.

Data	The Data table was initially split into Measured_Data and Computed_Data. This complicates the model because it provides a split of values across two tables. We have revised this numerous times, and now consider one table to be a valid solution. Computed data will have "COMPUTED" as the device (which is linked via Devise_ID). In the case of sound speed data, there may be a valid device (e.g., XSV) or COMPUTED device. We have also added an UNKNOWN device for those data that we are unsure of origin. Finally, we added Replicate_ID as a counter for replicated measurements from the same device. Note that device does not indicate a unique device (i.e., serial number) but rather a model of device. Devices could be added for uniqueness, but this was not the initial intent. Also, COMPUTED could be separated into various computation methods if so desired.
Data_Package	Contains a description of a collection of data that was added or ingested to the database as a single unit. The package should have a name and associated metadata.
Data_Packages	This is the master table that identifies specific data assets or resources. The Asset_ID has RoleName of Cruise_ID in other tables. This type of structure allows the incorporation of non-cruise data sets into the data model. For example, if model output is included it would not technically be related to a cruise. In this case, the Asset_ID would increment for the model output and likely be stored in non-cruise related tables.
Data_Set	A data set is a sub component of data package. A data set is a logically grouped portion of data. A data set may be grouped based on recording instrument, parameter type, etc.
Device_Class_Detail	This table provides information on the details of a specific class of device. This is NOT information specific to a particular device. As an example, the coefficients that should be used to process an XBT cast for a specific XBT type (e.g., T5, T7) would be noted here. Then, the actual values used in the processing are noted in Value.
EX_Extent	Table containing extent ID used for defining horizontal, vertical and temporal extent of a data set. See ISO 19115 element #334.
EX_Geographic_Bounding_Box	The geographic bounds of the data set. The bounds are expressed as a box in latitude/longitude space. See ISO 19115 element #343.
EX_Geographic_Extent	A table for the geographic extent ID and type. See ISO 19115 element #339.
EX_Temporal_Extent	The temporal extent of the data set. This table only stores a relationship link to the actual time period as expressed in TM_Period. See ISO 19115 element #350.

EX_Vertical_Extent	The vertical extent of the data set. Vertical extent is actually related to the coordinate reference system. Although ISO 19115 allows for this, we have not included an explicit link in this table. See ISO 19115 element #354.
Feature_Area	Feature_Area specifies a classification, name and geometry of an Area. The Feature_Code of the Area can be used to link to the Feature_Code within the Measurement_Location table. In that way, the defined Area can be associated with a constructed profile. In this way, we could construct standard profiles for defined Areas. Measurement_Location would not have X or Y, as the horizontal spatial area is defined within Feature_Area.
Feature_Asset	Table which lists all Feature_ID values that apply to a specific Asset_ID. This list of Feature_ID values is then subdivided into one of many possible tables in the lower structure of the data model.
LI_Lineage_Sources	Contains information about the source data set used in creating the data specific by Lineage_ID. See ISO 19115 element #92.
Instantaneous_Point	An Arc Marine table that contains information on single space-time measurements or computed values.
JO_Cited_Responsible_Parties	Join table for linking a person or organization to a citable reference.
JO_Data_Set_Data_Package	Join table for data package and data set. One data package can contain multiple data sets.
JO_Lineage_Data_Set	A join table for linking the data set with the lineage record for that data set.
JO_Process_Step_Processors	Join table for people performing the processing, and the processing steps.
JO_Process_Steps_Lineage	Join table for lineage and processing steps.
LI_Lineage	This is the top level table for the Lineage series of tables. This table links the asset ID number to the Lineage ID. A single asset can have many lineage ID values, indicating a sequence of processing steps that were applied to the asset. See ISO 19115 element #82.
LI_Process_Steps	Contains a description of the specific processing step. This is information about an event or transformation in the life of a data set including the processing used to maintain the data set. See ISO 19115 element #86.
Logins	A legacy table from the REA Load Database. This table lists the usernames and passwords for the system.
LU_Arc_Marine_Themes	The thematic types as described in Arc Marine. The themes are divisions or natural groupings for the data. See Arc Marine book, page 10.
LU_Arc_Marine_Types	The Arc Marine data types, which include Marine Points, Marine Lines, Marine Areas, Marine Rasters/Grids/Meshes. See page 14-15 or Arc Marine book.
LU_Data_Set_Formats	Contains descriptions of the formats ingested into the REA production database.
LU_Geometry_Classes	The geometry classes are the basic foundation geometries. Examples include, point, line, area.

Measurement_Location	The spatial location of the measurement. The ARC Marine model has this table named "Measurement". The name in this implementation was changed to more clearly identify the content. Note that a unique Feature_ID defines all member records of a single feature. The Measurement_Location table contains the X,Y,Z points even though for profile data, this violates 3rd normal form (there are no nonkey attributes which determine other nonkey attributes - in the profile case, the Feature_ID and Feature_Class determine the X and Y location values).
Measuring_Device	A description of all devices that could be used in the data collection activity. For measured data, it is typical to have a device description. However, for historic data the device related metadata may not exist. In this case, UNKNOWN is used as the device.
Mesh	This table defines the size of each mesh (i.e., each grid). The table provides the total number of points in the I,J,K directions (i.e., the X,Y,Z) for the mesh.
Mesh_Point	This table describes every point in the mesh, for every mesh, in terms of a unique Feature_ID number. This Feature_ID number is then used through the other model related tables to track this particular mesh/cell within the model.
Parameter	Contains all the parameters used in the Data table.
Profile_Notes	Any notes that were collected and pertain to the particular profile.
Quality_Flag	Contains the quality flags for the data values. All quality flags are listed in this table.
RS_Identifier	The reference system identifier. See ISO 19115 element #208.
Scalar_Quantity	This table defines the scalar quantities that are output from the model. Each quantity is time stamped and Feature_ID stamped.
Scientist	The names of all scientists that have been associated with data collection activities.
Scientist_On_Cruise	The names of the scientists that were involved in a particular cruise.
Series	The Series table is simply a means to collect together a group of Feature_IDs. The uniqueness of the Series_ID is maintained in the Series table. The table allows multiple Feature_IDs to be assigned to a single Series_ID. This allows the set of Features (i.e., set of Feature_IDs) to be grouped together to represent a line, or area.
Ship	The names of all ships associated with any data collection activity.
Ships_On_Cruise	The names of ships that took part in particular cruises.
Survey_Info	Survey_Info contains information on the survey. A survey is a data collection activity. Along any specific track, there can be multiple surveys. This is because multiple data collection activities can occur along a single track. For example, you may be running a sounder along a track while also dropping XBTs. These are considered two different surveys that occur along a single track. Note that relationships to the table do not require a Survey_ID be present (ie, nulls are allowed). This means the user DOES NOT have to specify a Survey before allowing data to be entered into the PDB.
Survey_Key	This table associates unique Survey_ID values to unique Feature_ID values. Thus, the table provides a joining of multiple features (i.e., multiple Feature_IDs) to a single survey.

TM_Period	The time period of the data set. This table contains a start and end date/time. The start and end should be expressed in ISO 8601 compliant form. However, this is not enforced at the field level due to the complexity of the ISO 8601 variations.
Track	A track is considered a line along which the ship moves. There may, or may not, be data collected during the transit of a ship. Likewise, there may, or may not, be data collected along a track. A track does not have to be a single line segment. Multiple segments can be included in a single track.
Transmission_Loss	Contains metadata associated with the transmission loss data. This includes much of the data that would have been in the header of the shallow water database files.
Valid_Classifications	A list of valid combinations of geometry classes and Arc Marine data types.
Vector_Quantity	This table defines the vector quantities that are output from the model. Each vector must be resolved into its components. Each quantity is time stamped and Feature_ID stamped.
Vehicle	This table describes the vehicle on which a measuring device is attached. A single vehicle can contain more than one measuring device. For example, a towed vehicle could have separate devices measuring temperature and pressure; a Remotely Operated Vehicle could carry an assortment of measuring devices; a marine mammal could be tagged and thus carry multiple measuring devices.
XBT	Contains attributes that are particular to the XBTProfile feature class. These attributes deal with assumed data values and processing details.

This page intentionally left blank.

Annex C REA production database table names, field names, field comments

Table 7: The PDB table, column and column comment fields in the data model.

Table Name	Column Name	Column Comment
Ambient_Noise	Approximate_Depth	An approximate depth (m) at the site.
	Feature_ID	An ID which is unique in the Feature_Asset table.
	Location_Comment	A general name for the location of the site.
	Shipping_Comment	A general comment on the amount of shipping that was taking place at the time of the experiment.
	Sediment_Comment	A general comment on the type of sediment at the site.
Area_Characteristic	Name	This is the name of the characteristic that is in the record.
	Value	This is intended to be a linked object or perhaps an image. The image could be something like the typical shapes of profiles for the specific region (i.e., Area).
	Feature_ID	The unique Feature identifier for the area characteristic.
	Characteristic_ID	A unique identifier for each individual characteristic that pertains to the Feature.
Mesh_Asset	Mesh_ID	The ID number associated with the mesh. This tells us which mesh the particular cell belongs to. This is a unique ID for the total mesh.
	Asset_ID	A sequential and unique number for each data asset.
CI_Citation	Edition	The version of the cited reference. ISO 19115, #363.
	Collective_Title	The title associated with the series or collective body of work. ISO 19115, #371.
	Title	The name by which the cited resource is known. ISO 19115, #360.
	Presentation_Form	The mode in which the resource is represented. ISO 19115, #368.

	Reference_Date	The date of the cited reference. ISO 19115, #362.
	Other_Citation_Details	Other information required to complete the citation that is not included elsewhere. ISO 19115, #370.
	ISSN	International Standard Serial Number. ISO 19115, #373.
	ISBN	International Standard Book Number. ISO 19115, #372.
	Edition_Date	Date of the edition. ISO 19115, #364.
	Citation_ID	A standardize resource reference. ISO 19115, #359.
	Alternate_Title	A short name or other language name by which the cited information is known. Example: WOA94 for World Ocean Atlas 94. ISO 19115, #361.
CI_Responsible_Party	ContactID	The unique ID assigned to the contact. This could be used to extend the model to include ISO 19115 element #387.
	PositionName	The role or position of the responsible party. ISO 19115, #377.
	IndividualName	The name of the responsible person. Surname, given name, title separated by delimiter. ISO 19115, #375.
	OrganizationName	The name of the responsible organization. ISO 19115, #376.
	Role	The function performed by the responsible party. ISO 19115, #379.
	RespPartyID	A unique identifier for the responsible party.
Cruise	Purpose	This is the reason the cruise is taking place. This might include the goals of the cruise.
	Name	This is a name that is associated with a cruise. The name could be related to the particular experiment, or perhaps the collaboration that involves the cruise (e.g., SAX04, NUW).
	Description	This is a broad and general description of the cruise.
	Start_Date	Start date of the cruise. When it first left port.
	End_Date	End date of the cruise.
	Status	Describes the current state of the cruise.

	Code	A user defined string for the cruise. For a typical DRDC cruise or field trial, this should be the cruise number or field trial number.
	Cruise_ID	A sequential and unique number for each data asset.
Cruise_Notes	Note_Counter	Sequential counter for the notes.
	Cruise_ID	A sequential and unique number for each data asset.
	Note	A general note field for the cruise.
CTD	CTD_Type_Code	Indicates the type of CTD being used. e.g., Seabird, Moving Vessel Profiler CTD
	Feature_ID	A ID which is unique in the Feature_Asset table.
Data	Data_Value	The value associated with the parameter and the device from which t was measured.
	Device_ID	The numeric used to group items based on their origin. Typically, the Device_ID indicates a numeric code for a particular class of devices. This type of grouping allows for multiple instruments of the same instrument code. For example, a hydrophone usually is used in multiples. So, an individual operation could use many hydrophones. We need a method of identifying the multiplicity of the instrument 'hydrophone'.
	Replicate_ID	Sequential counter for replicate values.
	Measurement_ID	A unique identifier that is used to relate the measured value to the measurement location.
	Parameter_ID	A unique identifier for the parameter property set.
	Quality_Flag	A character based flag that describes a quality condition on the data value.
Data_Package	Metadata_File	The filename corresponding to the metadata content.
	Data_Package_ID	A sequential and unique number for each data asset.
	Data_Package_Name	A sequential ID for each package. A package refers to a collection of data that is loaded into the database at one time. A package can have multiple data types or data sets.
Data_Packages	Data_Package_ID	A sequential and unique number for each data asset.
	Description	A description of the data asset.

Data_Set	Filename	The filename that was associated with the data set.
	Classification_ID	A unique sequential identifier for the joined geometry class, Arc Marine theme and the Arc Marine data type.
	Data_Set_ID	A unique sequential identifier for a particular data set. A data set is a subcomponent of a package. A data set has some type of naturally binding characteristic that allows us to consider it as a single set.
	Arc_Theme_ID	A sequential unique identifier for a particular Arc Marine theme. Themes include: Bathymetry and Backscatter; Mesh Volumes; Scientific Mesh; Survey Transects; Location Series Observations; Instantaneous Measured Points; Time Series Locations; Time Duration Features; Tracks and Cruises; Shorelines
	Data_Set_Format_ID	A sequential and unique identifier for a specific input format.
	Data_Set_Name	A name associated withy the data set.
Device_Class_Detail	Value	This is the value associated with the Descriptor property.
	Descriptor	This is a text description of a device property. This could be the name of a coefficient, the name associated with the terminal depth of the instrument, etc.
	Device_ID	The numeric used to group items in an instrument. This type of grouping allows for multiple instruments of the same instrument code. For example, a hydrophone usually is used in multiples. So, an individual operation could use many hydrophones. We need a method of identifying the multiplicity of the instrument 'hydrophone'.
EX_Extent	Extent_ID	A unique identifier for the extent.
EX_Geographic_Bounding_Box	West_Bound_Longitude	The western most coordinate of the limit of the data set extent, expressed in longitude in decimal degrees positive east. ISO 19115, #344.
	South_Bound_Latitude	The southern most coordinate of the limit of the data set extent, expressed in latitude in decimal degrees positive north. ISO 19115, #346.
	North_Bound_Latitude	The northern most coordinate of the limit of the data set extent, expressed in latitude in decimal degrees positive north. ISO 19115, #347.
	East_Bound_Longitude	The eastern most coordinate of the limit of the data set extent, expressed in longitude in decimal degrees positive east. ISO 19115, #345.

	Geoext_ID	A unique identifier for the geographic extent.
	Geobnbox_ID	A unique identifier for the geographic bounding box.
EX_Geographic_Extent	Extent_Type_Code	An indication of whether the bounding polygon encompasses an area covered by the data or an area where data is not present. ISO 19115, #340.
	Geoext_ID	A unique identifier for the geographic extent.
	Extent_ID	A unique identifier for the extent.
EX_Temporal_Extent	Tempext_ID	A unique identifier for the temporal extent.
	Extent_ID	A unique identifier for the extent.
EX_Vertical_Extent		The unique ID for the extent.
	Minimum_Value	The lowest vertical extent contained in the data set. ISO 19115, #355.
	Maximum_Value	The highest vertical extent contained in the data set. ISO 19115, #356.
	Vertext_ID	A unique ID for the vertical extent information.
Feature_Area	Feature_ID	The identifier for the Feature Area. Each area is uniquely identified by this Feature ID.
	Geom	The geometry of the area. In POSTGIS this would be a multi-polygon.
	Feature_Class	For the Feature_Area table, the Feature_Code identifies a general class of feature areas. For operation areas, this will be OPAREA. For general division of the ocean into areas (e.g., in the Banks 1950 report), we will have to invent a name which applies to these general areas (e.g., BANKS1950)
	Feature_Code	Area_Name is a common name assigned to the Feature_Area. For Operation Areas, this name might be ALPHA, CHARLIE1, etc.
Feature_Asset	Feature_ID	The identifier for this asset.
	Feature_Class	Identifies the classification of the Feature. Present classifications are: IP: Instantaneous Point FA: Feature Area
	Data_Package_ID	A sequential and unique number for each data asset.

Instantaneous_Point	Station_ID	The station number as defined by the original cruise documentation or drop sequence.
	Date_Time	The date and time associated with the value. Note that this is NULLable. This is because some values do not have a specific date and time. For example, temperature min and max profiles
	Feature_Code	Feature_Code is a subdivision of Feature_Class. Feature_Code is a string that describes the specific type of Feature_Class.
	Point_Type	Defines a subtype as one of the following: 1: Instant (default value) 2: Sounding 3: Survey 4: LocationSeries Note that LocationSeries is used to describe a transmission loss data set.
	Survey_ID	A unique identifier for a survey. A survey is considered as a collection of data from a specific area. A survey collects data at multiple locations. Multiple surveys can be associated with a single track. This is because along a single track, you could have multiple data collection activities going on.
	Z_Value	A single depth value for the point.
	Cruise_ID	A sequential and unique number for each data asset.
	Series_ID	A key field for relating this table to a Feature_Class.
	Feature_ID	An ID which is unique in the Feature_Asset table.
JO_Cited_Responsible_Parties	RespPartyID	A unique identifier for the responsible party.
	Citation_ID	A standardize resource reference. ISO 19115, #359.
JO_Data_Set_Data_Package	Data_Set_ID	A unique sequential identifier for a particular data set. A data set is a subcomponent of a package. A data set has some type of naturally binding characteristic that allows us to consider it as a single set.
	Data_Package_ID	A sequential and unique number for each data asset.

JO_Lineage_Data_Set	Data_Set_ID	A unique sequential identifier for a particular data set. A data set is a subcomponent of a package. A data set has some type of naturally binding characteristic that allows us to consider it as a single set.
	Lineage_Source_ID	A unique identifier for the source data. ISO 19115, #92.
JO_Process_Step_Processors	RespPartyID	A unique identifier for the responsible party.
JO_Process_Steps_Lineage	ProcStep_ID	A unique identifier for the processing step.
	Lineage_ID	A unique identifier for the change in lineage.
LI_Lineage	Lineage_ID	A unique identifier for the change in lineage.
	Statement	A statement which describes the change in lineage. This statement applies to the lineage for the entire data collection.
LI_Lineage_Sources	Identifier_ID	Unique number assigned to a reference system.
	Scale_Denominator	The denominator of the representative fraction on a source map. ISO 19115, #94.
	Lineage_Source_ID	A unique identifier for the source data. ISO 19115, #92.
	Extent_ID	A unique identifier for the extent.
	Lineage_ID	A unique identifier for the change in lineage. This is also a foreign key to the parent lineage record.
	Description	A statement which describes the source data. ISO 19115, #93.
	Citation_ID	A standardize resource reference. ISO 19115, #359.
LI_Process_Steps	Rationale	The requirement or purpose of the processing step. ISO 19115, #88.
	Date_Time	The date and time at which the processing step was applied. ISO 19115, #89.
	ProcStep_ID	A unique identifier for the processing step.
	Description	A description of the event including related parameters or tolerances. ISO 19115, #87.
Logins	Password	The password for the user of the database.
	ID	A unique ID for the user of the database.
	Username	A name for the user of the database.

LU_Arc_Marine_Themes	Description	The description of the Arc Marine theme.
	Arc_Theme_ID	A sequential unique identifier for a particular Arc Marine theme. Themes include: Bathymetry and Backscatter; Mesh Volumes; Scientific Mesh; Survey Transects; Location Series Observations; Instantaneous Measured Points; Time Series Locations; Time Duration Features; Tracks and Cruises; Shorelines
LU_Arc_Marine_Types	Arc_Marine_Type_ID	The sequential unique identifier for the GIS marine data type. These include: Marine Points; Marine Lines; Marine Areas; Marine Rasters/Grids/Meshes
	Description	A description of the marine data type.
LU_Data_Set_Formats	Data_Set_Format_ID	A sequential and unique identifier for a specific input format.
	Description	A description of the format.
LU_Geometry_Classes	Geometry_Class_ID	A unique sequential identifier for the geometry. These include: Feature Point Instantaneous Point -> Instant Instantaneous Point -> Survey Instantaneous Point -> Sounding Instantaneous Point -> Location Series Time Series Point -> Time Series Profile Line Time Duration Line -> Track Feature Line Shoreline Feature Area Time Duration Area Regularly Interpolated Surface Irregularly Interpolated Surface Mesh Volume Animation/Video/Movie
	Description	A description of the geometry class.

Measurement_Location	X_Y_Position_Code	The XY position will have an associated code to identify characteristics of the position. For example, some positions may be data while others are interpolated from nearby position fixes. This field will store the codes indicating the source of the position.
	Geom	The geometry of the point. In POSTGIS this would be a POINT.
	Feature_Code	Feature_Code is a subdivision of Feature_Class. Feature_Code is a string that describes the specific type of Feature_Class. Possible values for Feature_Code: XBT; CTD; AMBIENTNOISE; TRANLOSS; SEDIMENT; etc.
	Measurement_ID	A unique identifier that is used to relate the measured value to the measurement location.
	Feature_ID	A ID which is unique in the Feature_Asset table.
Measuring_Device	Description	The value associated with the name.
	Device_ID	The numeric used to group items in an instrument. This type of grouping allows for multiple instruments of the same instrument code. For example, a hydrophone usually is used in multiples. So, an individual operation could use many hydrophones. We need a method of identifying the multiplicity of the instrument 'hydrophone'.
	Name	The name of a particular item. In the inst_code="HP" example, the name field could be "HP Number", "HP Depth" or "HP Position". Unknown: refers to an unknown device Computed: Refers to a value obtained from a computation.
	Vehicle_ID	A unique number that identifies the vehicle used during data collection along this track. This is used in a relationship to the Vehicle Table.
Mesh	No_Of_Points_K	The total number of points (i.e., cells) in the K or Z direction of the mesh.
	No_Of_Points_J	The total number of points (i.e., cells) in the J or Y direction of the mesh.

	No_Of_Points_I	The total number of points (i.e., cells) in the I or X direction of the mesh.
	Total_Points	The total number of active points in the model domain. Note that this is active or modelled points. This is not simply the total number of I*J*K points.
	Mesh_ID	The ID number associated with the mesh. This tells us which mesh the particular cell belongs to. This is a unique ID for the total mesh.
	Dimension	Identifies if the mesh is one, two or three dimensional. Possible values for this field are: POINT; PLANE; VOLUME
Mesh_Point	Point_Type	Used to indicate if the mesh is regular or irregular. GridPoint indicates a regular box-type grid; NodePoint indicates an irregular grid common in finite element models.
	Feature_ID	This Feature_ID is a unique number for a unique cell within a particular mesh. This ID is used to track the cell output throughout these mesh tables.
	Geom	The latitude, longitude, depth of the particular cell's centre position.
	K_Position	The K (Z) position of the cell in the mesh. The convention will be that the first cell is number 1 (i.e., there is no 0 cell).
	J_Position	The J (Y) position of the cell in the mesh. The convention will be that the first cell is number 1 (i.e., there is no 0 cell).
	Mesh_ID	The ID number associated with the mesh. This tells us which mesh the particular cell belongs to. This is a unique ID for the total mesh.
	I_Position	The I (X) position of the cell in the mesh. The convention will be that the first cell is number 1 (i.e., there is no 0 cell).

Parameter	Name	This name is a code that identifies a specific data type being measured or reported on in the database. Note that this is a character code for the parameter. Using character codes it is much easier to identify errors when viewing the content of the database. For example, if the coding used numbers to represent data (e.g., 4,33.6) it is more difficult to visually recognize as error (e.g., WATERTEMP, 33.6). Also, this name and code can be used to describe non-measured values. For example, minimum/maximum envelopes for temperature can be given a unique parameter ID number, and associated Name and Description.
	Units	The unit associated with the parameter. Note that the same parameter name may have more than one unit. Not all parameters will have an associated unit (e.g., salinity).
	Quantity	The type of parameter. Arc Marine uses three types: 1 - other 2 - scalar 3 - vector
	Parameter_ID	A unique identifier for the parameter name.
	Description	A description of the parameter.
Position_Code	X_Y_Position_Code	The XY position will have an associated code to identify characteristics of the position. For example, some positions may be data while others are interpolated from nearby position fixes. This field will store the codes indicating the source of the position.
	Description	A description of the X_Y_Position_Code.
Profile_Notes	Station_ID	The station number from the cruise.
	Cruise_ID	A sequential and unique number for each data asset.
	Note	A descriptive note for this particular cruise/station combination.
Quality_Flag	Short_Description	A short description of what the quality flag means.

	Secondary_Flag	A secondary flag which means the equivalent of the quality flag, but in another quality flagging system. For example, this may be the quality flagging equivalent in the Integrated Science Data Management (formerly the Marine Environmental Data Service, or MEDS) system. Note that this field could also act to combine our quality flags. For example, we may have multiple flags which indicate the cause of the quality qualification, while the secondary flag indicates the impact on the data.
	Long_Description	A long description of what the quality flag means.
	Quality_Flag	A character based flag that describes a quality condition on the data value.
RS_Identifier	Identifier_ID	Unique number assigned to a reference system.
	Version	The cited reference system version, e.g. 6.13. ISO 19115, #208.2
	Code	The alphanumeric value identifying the source reference system, e.g. epsg.
	Code_Space	Identifier/namespace of the system in which the code is valid, e.g. http://www.epsg.org/databases/epsgv6_13.zip ISO 19115, #208.1
	Citation_ID	A standardize resource reference. ISO 19115, #359.
Scalar_Quantity	Feature_ID	This Feature_ID is a unique number for a unique cell within a particular mesh. This ID is used to track the cell output throughout these mesh tables.
	Data_Value	The value of the scalar.
	Date_Time	The date and time associated with the quantity.
	Parameter_ID	A unique identifier for the parameter name.
Scientist	Scientist_Name	The name of scientists. The format should be first name last name. e.g. John Smith
	Scientist_ID	Unique identifier for a scientist.
Scientist_On_Cruise	Scientist_Name	The name of scientists. The format should be first name last name. e.g. John Smith
	Scientist_Counter	A sequential counter for the scientists involved in the cruise.
	Cruise_ID	A sequential and unique number for each data asset.

Series	Series_ID	A key field for relating this table to a Feature_Class.
	Description	A description of the series.
Ship	Ship_Name	Name of the ship involved in the cruise.
	Ship_ID	Unique identifier for a ship.
Ships_On_Cruise	Ship_Name	Name of the ship involved in the cruise.
	Ship_Counter	A sequential counter for the ships involved in the cruise.
	Cruise_ID	A sequential and unique number for each data asset.
Survey_Info	Device_ID	The numeric used to group items in an instrument. This type of grouping allows for multiple instruments of the same instrument code. For example, a hydrophone usually is used in multiples. So, an individual operation could use many hydrophones. We need a method of identifying the multiplicity of the instrument 'hydrophone'.
	Start_Date_Time	Start date of the survey.
	End_Date_Time	End date of the survey.
	Track_ID	A unique identifier for a track. Recall that multiple surveys can occur on a single track. e.g. a single track between point A and B could have an XBT survey, a NADAS survey and a towed array survey; all occurring along the same track.
	Survey_ID	A unique identifier for a survey. A survey is considered as a collection of data from a specific area. A survey collects data at multiple locations. Multiple surveys can be associated with a single track.
	Description	A general description of the survey.
Survey_Key	Survey_ID	A unique identifier for a survey. A survey is considered as a collection of data from a specific area. A survey collects data at multiple locations. Multiple surveys can be associated with a single track.
	Feature_ID	A geodatabase wide unique identifier. It is also the Primary Key field for this table.
TM_Period	Start_Date	The starting date of the data being described by this particular lineage record. We recognize that starting dates for different data sources will vary. We suggest ISO 860-1 compliant dates, but recognize the fact that dates may be truncated.

	End_Date	The ending date of the data being described by this particular lineage record. We recognize that ending dates for different data sources will vary. We suggest ISO 860-1 compliant dates, but recognize the fact that dates may be truncated.
	TM_Period_ID	A unique sequential identifier for the time period of the lineage record.
	Tempext_ID	A unique identifier for the temporal extent.
Track	Description	Used to describe the specifics of the Track itself.
	Method	Used to describe specific methods used for this track.
	Name	Tracks that are commonly used for data collection may have been assigned a common name by which the track is known. This field is used to store that name. e.g., Hfx Line. To have a name, the Tracks are typically repeated over many different cruises.
	Local_Desc	The Track may have an official name in the Name field; but may also be known by a local name.
	Geom	The geometry of the track. In POSTGIS this would be a LINESTRING.
	Start_Date_Time	The date and time when the track was started.
	End_Date_Time	The date and time when the track was completed.
	Vehicle_ID	A unique number that identifies the vehicle used during data collection along this track. This is used in a relationship to the Vehicle Table.
	Track_ID	A unique number that identifies the Track.
	Feature_Code	A string that describes in some way, the feature contained in the current table. In the REA PDB, this is a letter combination that indicates the kind of feature described by the current row.
	Cruise_ID	A sequential and unique number for each data asset.
	Feature_ID	A geodatabase wide unique identifier. It is also the Primary Key field for this table.
Transmission_Loss	Station_Depth	The total water depth in metres (m) at the site of the mooring.
	No_Of_Shots	The total number of shots used on the run.
	No_Of_Frequencies	The total number of frequencies used to bin the data from the run.

	Sea_State	The sea state description. This is a coded value of 0-6, based on the international sea state scale.
	Maximum_Range	The range (in km) to the farthest shot position.
	Minimum_Range	The range (in km) to the closest shot position.
	Minimum_Run_Depth	The minimum water depth in metres (m) along the run line.
	Maximum_Run_Depth	The maximum water depth in metres (m) along the run line.
	Line_Direction	The direction of the run relative to the mooring. e.g. OPENING; CLOSING; CIRCULAR.
	Source_Depth	The depth in metres (m) at which the sound source is activated.
	Creation_Date_Time	The date and time when the original file was created.
	Edit_Date_Time	The date and time of the most recent edit to the input file.
	Bearing	The bearing in degrees true, toward which the ship or aircraft was moving during the transmission loss experiment.
	Band_Width	The band width of the frequency bins.
	Bottom_Type	A single word description of the bottom type along the run line. e.g., sand, clay, gravel, etc.
	Feature_ID	A ID which is unique in the Feature_Asset table.
	Sound_Description	A general description of the sound velocity profile.
Valid_Classifications	Arc_Marine_Type_ID	The sequential unique identifier for the GIS marine data type. These include: Marine Points; Marine Lines; Marine Areas; Marine Rasters/Grids/Meshes
	Classification_ID	A unique sequential identifier for the joined geometry class, Arc Marine theme and the Arc Marine data type.

	Geometry_Class_ID	A unique sequential identifier for the geometry. These include: Feature Point Instantaneous Point -> Instant Instantaneous Point -> Survey Instantaneous Point -> Sounding Instantaneous Point -> Location Series Time Series Point -> Time Series Profile Line Time Duration Line -> Track Feature Line Shoreline Feature Area Time Duration Area Regularly Interpolated Surface Irregularly Interpolated Surface Mesh Volume Animation/Video/Movie
Vector_Quantity	Feature_ID	This FeatureID is a unique number for a unique cell within a particular mesh. This ID is used to track the cell output throughout these mesh tables.
	Z_Component	The z-component value of the vector. All component values are the magnitude TOWARDS that direction (i.e., the oceanographic method of vector direction).
	Y_Component	The y-component value of the vector. All component values are the magnitude TOWARDS that direction (i.e., the oceanographic method of vector direction).
	X_Component	The x-component value of the vector. All component values are the magnitude TOWARDS that direction (i.e., the oceanographic method of vector direction).
	Date_Time	The date and time associated with the quantity.
	Parameter_ID	A unique identifier for the parameter name.
Vehicle	Name	The name associated with the vehicle.
	Vehicle_ID	A unique number that identifies the vehicle used during data collection along this track. This is used in a relationship to the Vehicle Table.
	Category	A descriptive category within which the vehicle falls. Examples might be ROV; array;

XBT	Sounding	The water depth in metres (m) at the point of XBT deployment. On QUEST, this value is typically obtained from the NADAS display.
	Serial_Number	The serial number of the instrument. At the time of model design, the content of Serial_Number is highly suspicious. However, if the field is present and we can get higher quality information input by the person doing the XBT drop, then the data quality will improve.
	Assumed_Salinity	The salinity used in the calculation to obtain sound speed from a temperature profile. Typically this salinity is a simple estimate.
	Deck_Unit	The deck unit used for recording the temperature profile from an XBT. Typical values would be MK12, MK21.
	Coefficient_B	The "b" coefficient for the XBT fall rate equation.
	Coefficient_A	The "a" coefficient for the XBT fall rate equation.
	Surface_Temperature	An estimate of the surface temperature value oin degrees C.
	Equation	Allowing 50 characters, we hope to allow the option of expressing the equations in a text form (e.g., T-5: z(t) = 6.54071t - 0.0018691t^2 where z(t) is depth in meters at time, t, in seconds).
	Feature_ID	An ID which is unique in the Feature_Asset table.

This page intentionally left blank.

Annex D Validation Lists

Table 8: Listing of validation codes used in the indicated tables and fields.

Table Name	Field Name	Validation List Name	Validation Values
Device_Class_Detail	Descriptor	Device_Class_Detail_List	XBTEquation
			XBTCoefficientA
			XBTCoefficientB
			XBTTerminalDepth
Instantaneous_Point	Feature_Code	Feature_Code_List	XBT
			TRANLOSS
			CTD
			FFCPT
			NADAS
			SEDIMENT
			AMBIENTNOISE
Mesh	Dimension	Mesh_Dimension	POINT
			PLANE
			VOLUME
Instantaneous_Point	Point_Type	Point_Type_List	1
			2
			3
			4
Parameter	Quantity	Parameter_Quantity	1
			2
			3
Transmission_Loss	Sea_State	Sea_State_List	0
			1
			2
			3
			4
			5
			6

Transmission_Loss	Line_Direction	TL_Line_Direction	OPENING CLOSING CIRCULAR
Measurement_Location	Feature_Code	Feature_Code_List	XBT TRANLOSS CTD FFCPT NADAS SEDIMENT AMBIENTNOISE

Annex E NADAS Data Stream

The Non Acoustic Data Acquisition System (NADAS) is a set of sensors and associated data streams that flow to a central computer system. The NADAS was constructed in the early 1980s on the premise that a flexible data acquisition system was required to collect environmental data while onboard the research vessel CFAV Quest. The earliest report of the NADAS system that could be located is by Guptill (1994).

Each sensor connected to the NADAS provides data to a central processing computer. The processing computer constructs a standardized output based on the sensor inputs and operator defined time intervals. This processing computer then redistributes the standardized output over the ship's serial network. In that way, the data stream is made available to other remote displays and remote computers via the serial connection. The system is scalable in the sense that multiple sensors can be added via multiple ports, without impacting the design.

The temporal operation of the system is depicted in Figure 21. A main processing loop (shown as the outer most loop) runs continuously (i.e., it starts immediately after it completes). Functionally, this loop controls the sequencing of the interior loops.

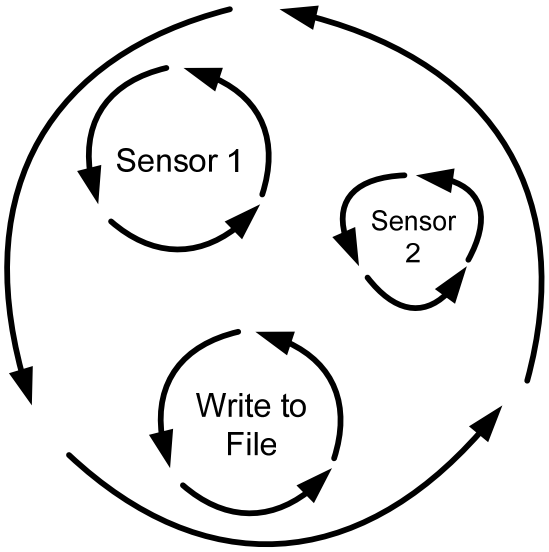

Figure 21: The NADAS system is represented as numerous processing loops. The main loop, shown here as the outer loop, runs continuously. It restarts immediately upon finishing. Each sensor has its own processing loop, shown here as interior sensor loops. Sensor loops are on independent timing loops. A special interior loop is denoted as the "Write to File" loop.

Most interior loops are sensor specific, as indicated in Figure 21 by the loops surrounding Sensor 1 and Sensor 2. These sensor loops have timing controls for some of the processing within the loop (see Figure 22). The processing, as represented here by the sensor loop, includes the functions to read the incoming data stream from the specifically identified port. The processing includes parsing the incoming data stream and assigning the parsed data to variables that are available throughout the processing environment (i.e., global variables). These variables may be used within the processing loop to construct a NADAS record. As well, the global variables may be used by other processing loops.

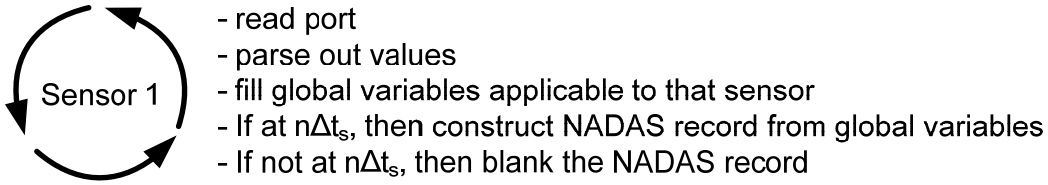

Figure 22: Generalized functionality of a NADAS processing loop. The Δt_s is the time interval for constructing an output record for that specific sensor. This is referred to as the construction time interval.

The construction of the NADAS record (example shown in Figure 23) is dependent on the timing control set by the user. The timing control establishes a construction time interval for creating the NADAS record particular to that sensor. This construction interval is defined by the user and is indicated in Figure 22 as Δt_s where s indicates a sensor specific subscript. At a time equal to any integer number of Δt_s, the NADAS record is constructed by the processing loop. This record is placed in a separate global variable. For processing times which are not integer intervals of Δt_s, the processing loop blanks out the NADAS global variable which contains the NADAS record output for that sensor. Here, "blanks out" refers to the empty string represented by "". The construction and emptying of the global variable for the NADAS record is represented in Figure 24 as a delta function which spikes at the Δt_s interval.

036,2004/05/29,04:28:20,02.5,031

Figure 23: An example NADAS record. This record has a NADAS code of "036". The code is followed by date and time in UTC. The 036 code indicates the record contains speed over ground (SOG, in knots) and course over ground (COG in degrees true). The value 2.5 is the speed, while the 031 is the course.

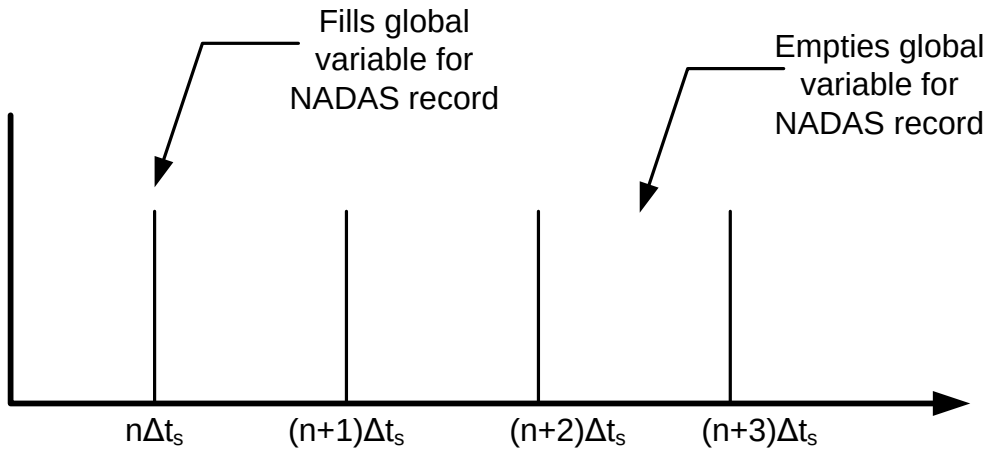

Figure 24: The construction of the NADAS record is based on the construction time interval established by the user. The construction is shown here as a delta function.

It is important to note that each sensor processing loop may or may not be successful in creating the NADAS record at the required time. The construction time interval is based on the UTC[2] value obtained from the GPZDA NMEA string. If the UTC seconds is divisible by the user defined construction time with a modulus of zero, then the NADAS record is constructed. Several factors may contribute to the NADAS record not being constructed at the expected time. Some of these factors include:

- sensor off line: In this case, no data stream would exist and the parsing would result in no values being obtained for the sensor specific global variables.
- missed data string: If the sensor data stream contains many different data strings, it is conceivable that when the processing loop acquires a data sample from the port, the sample does not include the specific data strings that are required for processing. In this case, the parsing would not find the particular data strings that it is looking for, and thus not fill the data global variables. If the time is appropriate for construction of the NADAS record, the record would contain empty values.
- missed data time: If the processing is slow, perhaps during early (i.e., 1980's) NADAS collection when slower computers were used, or perhaps due to numerous sensors in the input, the exact second that satisfies the $n\Delta t_s$ requirement may be missed.

[2] The NMEA 0183 standard indicates this is UTC on GPZDA. This is not GMT, and not GPS time.

The “Write to File” processing loop shown in Figure 21 indicates a special loop dedicated to the creation of the ASCII NADAS file. This file represents the output from the system. This loop has no timing control – it simply starts immediately after finishing. The loop writes all global variables corresponding to the NADAS record to an output file. Recall that the records may or may not contain data. The write proceeds even if no data are present in any particular record, as the write of an empty record simply produces no output (see Figure 25). At the end of the write loop, all global variables corresponding to the NADAS records are emptied.

If the “Write to File” processing loop writes multiple NADAS records, those records will all contain the same date/time stamp. This is because the records are constructed using the same global variable for the date/time, as determined from the GPZDA NMEA string. A set of NADAS records with the same date time strings are referred to as a NADAS record set.

NADAS Global Variable for record 013 – 013,2004/05/29,04:29:00,37:26.224N,012:23.613E,GPS
NADAS Global Variable for record 020 – 020,2004/05/29,04:29:00,82.38,37:26.224N,012:23.613E
NADAS Global Variable for record 030 – 030,2004/05/29,04:29:00,0.0,52.2
NADAS Global Variable for record 031 –
NADAS Global Variable for record 032 – 032,2004/05/29,04:29:00,276.40
.... [more exist]

Figure 25: The global variables are indicated for NADAS records 013, 020, 030, 031 and 032. Only the first three and fifth global variables contain data. At the write time, the NADAS record for 013, 020, 030 and 032 would appear in the output file. All other global variables are empty and thus do not contribute to the output.

E.1 The 013 NADAS record

The 013 NADAS record contains the GPS latitude and longitude values. For a collection system used during ship steaming, the positional information is critical for georeferencing the sensor data.

In the current NADAS system, the user specifies the construction time interval for each NADAS record, including the 013 record. The time interval is selectable from a drop down list containing the most widely used values (e.g., 10, 30, 60 seconds). However, the user may also specify this interval in a text box. This allows the user to specify any possible write interval to the resolution of one second.

E.1.1 NADAS software recommendation 1

This level of user specification means it is possible to specify a set of construction time intervals which result in the 013 record not being present in all record sets. The most obvious scenario for such a result would be specifying the 013 record at a 60 second interval with other records at 10 second intervals. In this scenario, interpolation is required to georeference the sensor data that was collected at 10 second intervals.

Having the system automatically set the 013 construction time interval to the lowest time interval specified, does not necessarily result in the 013 record being present in each record set. If Sensor 1 time interval is set to five seconds, while Sensor 2 time interval is 13 seconds, the minimum interval is not divisible into the maximum. In this case, if 013 were set to the minimum construction time, it would not exist in the Sensor 2 record set.

Perhaps a possible solution to 013 being present in every record set is to extend the drop down list box of allowable times to include 1, 2, 10, 30, 60, 120 seconds and to also remove the user defined time interval. With these restrictions, the 013 construction time could be set automatically to the minimum construction time interval for all sensors.

E.2 Number of 013 Records

The REA LDB version 3 beta was queried to determine the number of record blocks containing the 013 record. The _temp_nadas_lines_decoded_parts table was queried for distinct *thetimestamp* values, and 3,122,461 records were found. Then, the same table was queried to create a new table with 013 records and the corresponding *thetimestamp* value for the 013 record. The second table was queried for distinct *thetimestamp*, finding 2,613,958 records. The record ratio is 83.7 %; indicating that 83.7% of all record sets contain 013 coded lines.

E.2.1 NADAS recommendations for ingesting data into REA LDB

Since not all record sets will contain the georeference data available on record 013, it is important to develop a strategy for ingesting the NADAS data that accounts for missing georeference data. Our recommendations are as follows:

1. Use 013 record position for a record set when record 013 is present in the set

2. If record 013 does not exist in the record set, use record 020 for georeference position. Note that position data were added to record type 020 at about 1994.

3. If record 020 does not exist in the record set, determine the position based on interpolation using 013 or 020 records in the temporally closest record sets. Use interpolation when the temporal separation of the georeferenced record sets does not exceed 2 minutes in total span (i.e., between position fixes). Note that the 2 minute separation is arbitrary.

This page intentionally left blank.

Annex F Sediment Thickness Information

The following text is reproduced from the help file associated with sediment thickness data.

Total Sediment Thickness of the World's Oceans & Marginal Seas

A digital total sediment thickness database for the world's oceans and marginal seas is being compiled by the National Geophysical Data Center (NGDC). The data will be gridded with a grid spacing of 5 arc-minutes by 5 arc-minutes. Sediment thickness data were compiled from three principle sources: previously published isopach maps including Ludwig and Houtz [1979], Matthias et al. [1988], Divins and Rabinowitz [1990], and Hayes and LaBrecque [1991]; ocean drilling results, both from the Ocean Drilling Program (ODP) and the Deep Sea Drilling Project (DSDP); and seismic reflection profiles archived at NGDC as well as seismic data and isopach maps available as part of the IOC's Geological/Geophysical Atlas of the Pacific (GAPA) project.

The distribution of sediments in the oceans is controlled by five primary factors:

- Age of the underlying crust
- Tectonic history of the ocean crust
- Structural trends in basement
- Nature and location of sediment source, and
- Nature of the sedimentary processes delivering sediments to depocenters

The digital data were produced in the following manner. First, the contour maps were digitized. The digitization of the Pacific was performed by Greg Cole of Los Alamos National Laboratory, and for the Indian Ocean by Carol Stein of Northwestern University, the South Atlantic and Southern Ocean by Dennis Hayes of Lamont-Doherty Earth Observatory. The second step was to grid the data, the algorithm for "Gridding with Continuous Curvature Splines in Tension" of Smith and Wessel [1990] was used. The gridding was performed at NGDC.

The data values are in meters and represent the depth to acoustic basement. It should be noted that acoustic basement may not actually represent the base of the sediments. These data are intended to provide a minimum value for the thickness of the sediment in a particular geographic region.

Sediment thickness values represent an average over each 5-minute grid cell and are located at the center of each cell. The ASCII files contain the position of the center of each cell.

How to Cite These Data:

Divins, D.L., NGDC Total Sediment Thickness of the World's Oceans & Marginal Seas, Retrieved date goes here, http://www.ngdc.noaa.gov/mgg/sedthick/sedthick.html

Other References:

Divins, D.L., and B. Eakins, Total Sediment Thickness Map for the Southeast Pacific Ocean, edited by G.B. Udintsev, Intergovernmental Oceanographic Commission, International Geological-Geophysical Atlas of the Pacific Ocean, in preparation.

Divins, D.L., and P.D. Rabinowitz, Total sediment thickness map for the South Atlantic Ocean, in International Geological and Geophysical Atlas of the Atlantic and Pacific Oceans (GAPA), edited by G.B. Udintsev, pp. 147-148, Intergovernmental Oceanographic Comission, 1991.

Hayes, D.E., and J.L. LaBrecque, Sediment Isopachs: Circum-Antarctic to 30S, in Marine Geological and Geophysical Atlas of the Circum-Antarctic to 30S, edited by D.E. Hayes, pp. 29-33, American Geophys. Union, Washington, D.C., 1991.

Smith, W.H.F., and P. Wessel, Gridding with Continuous Curvature Splines in Tension, Geophysics, 55, 1990.

Ludwig, W.J., and R.E. Houtz, Isopach Map of the Sediments in the Pacific Ocean Basin, color map with text, Am. Assoc. Pet. Geol., Tulsa, OK., 1979.

Mathhias, P.K., P.D. Rabinowitz, and N. Dipiazza, Sediment Thickness map of the Indian Ocean, Map 505, Am. Assoc. Pet. Geol., Tulsa, OK., 1988.

List of symbols/abbreviations/acronyms/initialisms

ARP	Applied Research Project
BODC	British Oceanographic Data Centre
CF	Canadian Forces
CFAV	Canadian Forces Auxiliary Vessel
COG	course over ground
CRS	coordinate reference system
CTD	conductivity-temperature-depth
DBMS	database management system
DND	Department of National Defence
DREA	Defence Research Establishment Atlantic
DRDC	Defence Research & Development Canada
DRDKIM	Director Research and Development Knowledge and Information Management
DRP	Document Review Panel
DSDP	Deep Sea Drilling Project
EPSG	European Petroleum Survey Group
GAPA	Geophysical Atlas of the Pacific
GCRS	georeferenced coordinate reference system
GIS	geographic information system
GMT	Greenwich Mean Time
GPS	global positioning system
IE	Information Engineering
ISO	International Organization for Standardization
JC3IEDM	Joint Consultation, Command and Control Information Exchange Data Model
JDBC	Java database connectivity
LDB	load database
LGPL	Lesser General Public License
MIEM	Maritime Information Exchange Model
NAD	North American Datum
NAD27	North American Datum of 1927

NAD83	North American Datum of 1983
NADAS	Non Acoustic Data Acquisition System
NGDC	National Geophysical Data Centre
NIEM	National Information Exchange Model
NMEA	National Marine Electronics Association
ODBC	open database connectivity
ODP	Ocean Drilling Program
OGC	Open Geospatial Consortium
OGP	Oil & Gas Producers
PDB	production database
R&D	Research & Development
REA	rapid environmental assessment
REAS	rapid environmental assessment system
SQL	structure query language
SRID	spatial reference identifier
SWDB	Shallow Water Database
TM	Technical Memorandum
TSD	Temperature-salinity Dalhousie
UTC	universal time coordinated
WGS84	World Geodetic System of 1984
XBT	eXpendable BathyThermographs
XML	extensible markup language

Glossary

Arc Marine

the outline or framework for an oceanographic database. Arc Marine was developed by Oregon State University specifically for use with the ESRI Arc series of products. The outline does not specify details of the database structure, but rather outlines a broad structure to be extended for user specific needs.

attribute

the name of a specific field-like property contained in an entity in a data model.

coordinate reference system

a mathematical model used to define the ellipsoid for fitting to the Earth.

Database Management System (DBMS)

a software system that allows the creation of a database. The DBMS typically supports the SQL standard as a means to interact with the database.

database

a specific storage unit built from tables, fields, and relationships that meet specific needs of the user's data. A database is constructed within a DBMS.

data asset

the term used to include data package, data set and the REA PDB in its entirety.

data model

a blue-print like description for the design of a database. There are multiple types of data models including conceptual, logical and physical.

data package

a collection of data that are loaded into the REA PDB using a single load operation.

data set

a subcomponent of data package. The data set is a coherent unit of data that may be treated in a similar way. The data set division may be based on a single parameter type.

data type

a particular environmental data measurement category. Data types are specific measurement types such as temperature, salinity or transmission loss.

entity

an object in a data model that represents a table in a database.

Feature

a specific geospatial characteristic of the land- or ocean-scape. Features can be represents by points, lines or areas. Features may also change over time, such as being non-existent at one time and existent at another time.

Feature Code

a subdivision of a Feature. This code indicates a real-world subdivision of the Feature. E.g., a set of Features that are points, could be subdivided into points corresponding to XBT drop location, CTD cast locations, or transmission loss mooring locations.

Feature_ID

a unique integer identifier for a feature.

field

a specific location for data storage contained in a database table.

geodatabase

a database that has been spatially enabled through the use of geometry encodings for the positional information.

georeferenced coordinate reference system

the ellipsoid after being fitted to Earth data points. The fitted point may be from local or global measurements, thus producing either a local of global GCRS.

geometric type

the characteristics of a measured set of data. Geometric types include profiles, horizontal surfaces, etc. and represent a category of data.

Geometry

in GIS terminology, an encoded representation of a point, line or area. The encoding uses the geospatial positions of the point, line or area with the georeferenced coordinate reference system, transforming those positions into an encoded string. This encoding increases geospatial query processing speed.

key

an abbreviated form of ‘primary key’.

nonkey

refers to those fields in a database table that are not part of the primary key.

North American Datum (NAD)

a spatial reference frame used on the North American continent.

NADAS record set

a group of records in the NADAS output file that all have the same date/time stamp.

primary key

a single field or a set of fields in a database table, that provide a unique identifier for the row of data.

referential integrity

a term used in database management to indicate system enforced consistency between multiple data values.

spatial reference frame

the specification that relates a position measurement to the physical earth.

specification

A general description which has no limitations on the level of required documentation and no requirement for formal approval, publishing or governance by a broad community-based organisation. (based on Stocks, Graybeal, *et al.* (2009)).

standard

A set of documented rules which define the creation of something. These rules provide a combination of terminology (vocabularies), syntactical rules, format rules, and other requirements. Standards are approved, published and governed by a formal body or organization with broad community-based representation (international or national). (based on Stocks, Graybeal, *et al.* (2009)).

table

the object in a database which contains data.

theme

a contrived assembly of data with similar characteristics. In the Arc Marine data model, themes include: shorelines; tracks and cruises; time duration features; time series locations; instantaneous measured points; location series observations; survey transects; scientific mesh; mesh volumes; and bathymetry and backscatter.

World Geodetic System

a spatial reference frame used by the satellites that make up the Global Positional System.

This page intentionally left blank.

Distribution list

Document No.: DRDC Atlantic TM 2009-061

LIST PART 1: Internal Distribution by Centre

2 DRDC ATLANTIC LIBRARY FILE COPIES
3 DRDC ATLANTIC LIBRARY (SPARES)
1 SECTION HEAD MICS
1 SECTION HEAD US
1 TIM HAMMOND
1 PAUL HINES
1 LIESA LAPINSKI
1 ANDREW MACINNIS
1 MARK MCINTYRE
1 JOHN OSLER
1 SEAN WEBB
2 A. W. ISENOR (1 CD COPY, 1 HARD COPY)
4 TOBIAS SPEARS (2 CD COPIES, 2 HARD COPIES)

20 TOTAL LIST PART 1

LIST PART 2: External Distribution by DRDKIM

1 Library and Archives Canada
Atten: Military Archivist, Government Records Branch

1 DRDKIM

1 Senior Staff Officer MetOc
MetOc, Bldg D-201
CFB Halifax
Halifax, NS
B3K 5X5

1 Staff Officer Military Oceanography
MetOc, Bldg D-201
CFB Halifax
Halifax, NS
B3K 5X5

1 Peter Giles
GD Canada
31 Millbrook Avenue,
Dartmouth, NS
B2V 0A2

1 Ulrich Suesser
suesser.US@forces.gc.ca
PO Box 17000 Stn Forces
Victoria, BC
V9A 7N2

1 Raffaele Grasso
NURC, NATO Research Centre
Applied Research Department
Information Fusion and Knowledge Management Group
Viale S. Bartolomeo 400
19126 La Spezia, Italy

7 TOTAL LIST PART 2

27 TOTAL COPIES REQUIRED

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada – Atlantic
9 Grove Street
P.O. Box 1012
Dartmouth, Nova Scotia B2Y 3Z7

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Utilizing Arc Marine concepts for designing a geospatially enabled database to support rapid environmental assessment

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Isenor, A.W.; Spears, T.W.

5. DATE OF PUBLICATION (Month and year of publication of document.)

July 2009

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

174

6b. NO. OF REFS (Total cited in document.)

64

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Memorandum

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Atlantic
9 Grove Street
P.O. Box 1012
Dartmouth, Nova Scotia B2Y 3Z7

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

Projects 11CQ; 11CH; 11CN

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Atlantic TM 2009-061

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The design of the rapid environmental assessment (REA) database version 1 was completed under contract. The database was constructed in PostgreSQL, an open-source database management system. The REA database was primarily used for the storage of DRDC Atlantic environmental data. However, additional data sets from external sources were added for bathymetry and geological data. As use of the REA database increased, it became desirable to redesign the database to better serve the user community within DRDC Atlantic. The redesign effort focused on the use of widely used standards and specifications for oceanographic data and metadata management. The redesign created a complete data model in the ERwin data modelling software for a production level database. The data model is fully documented in terms of data type, comment fields and relationships between entities. The redesign effort also fully documented the mapping of data from the existing REA database to the redesigned production data model, thereby providing developers with a clear and concise progression plan. The redesign effort also identified considerable data in the existing REA database that is not required in the production database. Finally, a designed data classification scheme is used to develop a user exit strategy for accessing the external data sets. This negates the need to store external data sets within the redesigned database.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

rapid environmental assessment; data modelling; Arc Marine; PostgreSQL; geographic metadata; geographic information system; GIS; PostGIS; ISO 19115

This page intentionally left blank.

Defence R&D Canada

Canada's leader in defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

www.drdc-rddc.gc.ca